DEVELOPMENT OF A LOW-COST PORTABLE, DYNAMIC ARDUINO BASED LIGHT MEASURING INSTRUMENT FOR CCT AND ILLUMINANCE MEASUREMENT SUITABLE FOR DIFFERENT INDOOR & OUTDOOR APPLICATIONS

A THESIS

SUBMITTED ON THE PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING

IN

ILLUMINATION ENGINEERING

SUBMITTED BY:

SOHAM SAIN

EXAMINATION ROLL NO: M4ILN24005

REGISTRATION NO: 163498 of 2022-2023

UNDER THE SUPERVISION OF

Dr. SUDDHASATWA CHAKRABORTY

Associate Professor, Jadavpur University

Electrical Engineering Department

Faculty of Engineering and Technology

Jadavpur University

Kolkata - 700032

May, 2024

DEVELOPMENT OF A LOW-COST PORTABLE, DYNAMIC ARDUINO BASED LIGHT MEASURING INSTRUMENT FOR CCT AND ILLUMINANCE MEASUREMENT SUITABLE FOR DIFFERENT INDOOR & OUTDOOR APPLICATIONS

A THESIS

SUBMITTED ON THE PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING

IN

ILLUMINATION ENGINEERING

SUBMITTED BY:

SOHAM SAIN

EXAMINATION ROLL NO: M4ILN24005

REGISTRATION NO: 163498 of 2022-2023

UNDER THE SUPERVISION OF

Dr. SUDDHASATWA CHAKRABORTY

Associate Professor, Jadavpur University

Electrical Engineering Department

Faculty of Engineering and Technology

Jadavpur University

Kolkata - 700032

May, 2024

JADAVPUR UNIVERSITY

<u>DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS</u>

I hereby declare that the discussion entitled "DEVELOPMENT OF A LOW-COST PORTABLE, DYNAMIC ARDUINO BASED LIGHT MEASURING INSTRUMENT FOR CCT AND ILLUMINANCE MEASUREMENT SUITABLE FOR DIFFERENT INDOOR & OUTDOOR APPLICATIONS", being submitted by SOHAM SAIN towards the partial fulfillment of the degree of Master of Engineering in Illumination Engineering, in the Department of Electrical Engineering, is a Thesis work carried out, under the supervision of Dr. SUDDHASATWA CHAKRABORTY, and has not been submitted anywhere else.

This thesis contains literature survey and original research work by the undersigned candidate, as a part of the studies. All the information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by thesis rules and conduct, I have fully cited and referenced all material and results which are not original to this work.

NAME	**:**	**SOHAM SAIN**
EXAMINATION ROLL NO.	**:**	**M4ILN24005**
REGISTRATION NO.	**:**	**163498 OF 2022-2023**
THESIS TITLE	**:**	**Development Of a Low-Cost Portable, Dynamic Arduino Based Light Measuring Instrument for CCT and Illuminance Measurement Suitable for Different Indoor & Outdoor Applications**
SIGNATURE & DATE	**:**	

JADAVPUR UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

ELECTRICAL ENGINEERING DEPARTMENT

RECOMMENDATION CERTIFICATE

This is to certify that the thesis entitled “DEVELOPMENT OF A LOW-COST PORTABLE, DYNAMIC ARDUINO BASED LIGHT MEASURING INSTRUMENT FOR CCT AND ILLUMINANCE MEASUREMENT SUITABLE FOR DIFFERENT INDOOR & OUTDOOR APPLICATIONS” submitted by SOHAM SAIN, (Examination Roll No. **M4ILN24005**, Registration No. **163498** of **2022-2023**) of this University in partial fulfilment of requirements for the award of degree of Master Of Engineering in Illumination Engineering, Department of Electrical Engineering, is a genuine record of the work carried out by him under my guidance and supervision.

Dr. SUDDHASATWA CHAKRABORTY
(Thesis Supervisor)
Associate Professor
Electrical Engineering Department
Jadavpur University
Kolkata: 700032

PROF. BISWANATH ROY
Head of the Department
Electrical Engineering Department
Jadavpur University
Kolkata: 700032

PROF. DIPAK LAHA
Dean
Faculty of Engineering and Technology
Jadavpur University
Kolkata: 700032

JADAVPUR UNIVERSITY

188, Raja S.C. Mallick Road, Kolkata – 700032
Phone: (+91) 33 2457 2227, **Fax:** 033 2414 6973
Email: officer.placement@jadavpuruniversity.in
Website: www.jaduniv.edu.in

CERTIFICATE OF APPROVAL

This foregoing thesis is hereby approved as a creditable study in the area of Illumination Engineering, carried out and presented by **SOHAM SAIN**, in a manner of satisfactory warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is notified to be understood that by this approval, the undersigned do not necessarily endorse or approved the thesis only for the purpose for which it has been submitted.

FINAL EXAMINATION FOR EVALUATION OF THESIS

BOARD OF EXAMINERS

____

____

(Signature of Examiners)

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and indebtedness to **Dr. Suddhasatwa Chakraborty**, Associate Professor, Department of Electrical Engineering, Jadavpur University, Kolkata, without his mission and vision, this project would not have been possible.

I would like to acknowledge my sincere gratitude to Prof. (Dr.) **Biswanath Roy**, Head of the Department (HOD) of Electrical Engineering Department, Jadavpur University, Kolkata for providing me the opportunity to carry out my project work in Illumination Engineering Laboratory, Jadavpur University. Moreover, I would like to thank Prof. (Dr.) **Saswati Mazumdar**, former HOD of Electrical Engineering Department and Mrs. **Sangita Sahana**, assistant professor of illumination engineering; for providing me their valuable time and helpful suggestion.

I would like to convey my special thanks to **Mr. Pradip Pal** and **Mr. Samir Mandi** for helping me in pursuing my experiments in illumination engineering laboratory.

At last I want to convey my thanks to **Roni Jana** who helped me in the thesis work. Last but not least, I wish to convey my gratitude to my parents, whose love, teachings and support have brought me this far.

SOHAM SAIN

Jadavpur University

Kolkata: 700032

ABSTRACT

Measurement of Photometric Quantities such as illuminance, Red-Green-Blue (RGB) values of any light source and Correlated Color Temperature (CCT) calculation is so far dependent on Conventional instruments and methods. But as modern technology is switching towards robotics and automation thus measurement of Photometric Quantities can also be revolutionized through robotics. In this research paper we have integrated a RGB Color Sensor (TCS3200) and a Lux Sensor (BH1750) to measure Illuminance and R-G-B color content of any light source simultaneously. After that, we have developed an algorithm using Python Code to calculate CCT values of a light source from its R-G-B responses acquired from the sensors and governing an experimental look-up table. Moreover, a driving mechanism is introduced in this model to get the Real-Time photometric values at a particular grid point. Thus, this paper we throw light on the integration of Photometry and Robotics.

Keywords: RGB Sensor, RC Car, Arduino Uno, ESP32, Lux Measurement, CCT, Real Time Measurement, Bluetooth Control, Photometric Measurement, ESP32 Webserver

CONTENTS

CHAPTER 1: INTRODUCTION

As the population continues to grow and science and technology is evolving rapidly new technical innovations are needed for efficient and effective impact. Thus, we can tackle the challenges that faced by human beings in the new millennium. Sensors plays an important role in robotics and at the same time RGB sensor, Lux sensor etc. plays significant role in Photometry. The domain of Optical Sensors are crucial is robotics system. Optical sensors uses light to detect changes in the environment. Hence, we can bring a fine relation between robotics and photometry which will dictate the path of Modern Photometry.

But before moving to modern photometry, let's know briefly about what is the significance of light in photometry. Basically, **Light** is an electromagnetic radiation that travels in the form of wave and create sensation to human eye. Light encompasses a broad spectrum of electromagnetic waves, ranging from radio waves to gamma rays. But specifically wavelengths that ranges from 380 nm to 780 nm are called as Visible Spectrum. This spectrum range is responsible for creating the visual sensation to human eye. And the Peak Sensitivity of human eye lies at a wavelength of 555 nm. Beyond visible light, the spectrum extends into the infrared and ultraviolet regions, each with unique applications and implications in diverse scientific fields. Figure 1 shows the clear demarcation between all the electromagnetic spectrum ranges.

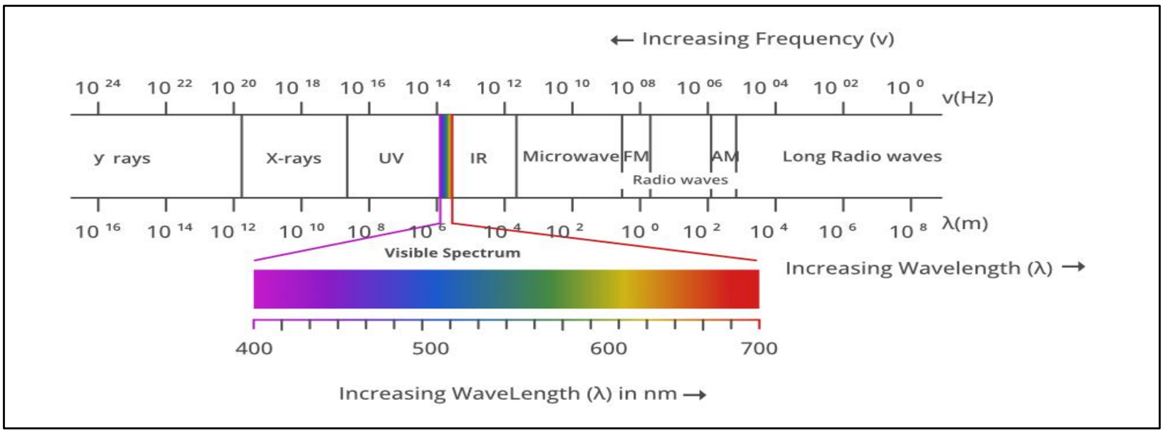


Figure 1: Electromagnetic Spectrum [2]

The electromagnetic spectrum [3] spans an enormous range of frequencies and wavelengths. If **C** is the speed of light (approximately 3 x 10^8 m/s in a vacuum) the relationship between frequency (**f**) and wavelength (**λ**) is described by the equation:

$$\boldsymbol{C = f \times \lambda}$$

Thus it is concluded from the equation that As frequency increases, wavelength decreases, and vice versa. Radio waves are the longest wavelengths and has the lowest frequencies within the electromagnetic spectrum. Microwaves have shorter wavelengths and higher frequencies compared to radio waves. They are widely used in household appliances like microwave ovens for cooking. Beyond microwaves, infrared radiation consists of longer wavelengths. Infrared is felt as heat and has applications in various fields, including night vision technology, thermal imaging, and industrial processes. Similarly after Visible region we have UV, X-ray and Gamma radiation region. Utilizing the phenomenon of light both Interior and exterior lighting is being designed for illumination. In most of the cases Avg. Illuminance (E_{avg}) and Uniformity (U) comes under consideration.

In this project, we detailed the process of calibrating an RGB sensor and a lux sensor, and then integrated these two sensors to measure both the illuminance and RGB levels of any light source. Utilizing these measurements, we developed a Python algorithm to calculate the Correlated Color Temperature (CCT) from the RGB responses. In the next stage of the project, we developed an electric vehicle (EV) operated via Bluetooth using L298N motor drivers. This robotic car integrates with the sensor setup and can be driven to specific grid points to measure the desired photometric quantities.

The lighting industry is rapidly evolving, not only in terms of energy efficiency but also in personalized human-centric lighting. Today, we can control and tune our smart LED bulbs to suit our needs through IoT technology. Similarly, robotics can offer various solutions for experiments and measurements in photometry, advancing both research and practical applications in the field.

1.1 ADVANCEMENT OF LIGHT MEASURING INSTRUMENTS:

The evolution of light measuring instruments [9] has paralleled advancements in lighting technology, becoming increasingly sophisticated and accurate. Let’s discuss about few significant measuring instruments from early days of lighting engineering to the recently developed sophisticated technologies.

- **Grease-Spot Photometer:** This simple device allowed for basic comparison of light sources but lacked precision and was highly dependent on human perception.
- **Selenium Photocells:** In the early 20th century, selenium photocells were used in some of the first electric light meters. These devices converted light into electrical signals, providing a more objective measurement of light intensity.

- **Photodiodes and Phototransistors:** Modern light meters often use photodiodes and phototransistors, which offer higher sensitivity and faster response times compared to earlier technologies. These components are integral to lux meters, colorimeters, and spectroradiometers.
- **Digital Lux Meters:** Digital lux meters (figure 14) provide accurate, easy-to-read measurements of illuminance, often with additional features such as data logging and connectivity to computers for further analysis.
- **Advanced Spectroradiometers:** These instruments measure the spectral power distribution of light sources, providing detailed information about color rendering, color temperature, and other critical parameters. They are essential tools in both research and industry for ensuring high-quality lighting.

1.2 IMPACT OF MEASUREMENT IN LIGHTING DESIGN:

The evolution of light measurement technology has revolutionized lighting design [11] leading to significant improvements in energy efficiency, visual comfort, and aesthetic appeal. Let's discuss about few of them in brief;

- **Energy Efficiency & Optimization:** Precise light measurements enable designers to optimize lighting systems for energy efficiency. By understanding the exact amount of light needed for different tasks and environments, unnecessary energy consumption can be minimized, leading to significant cost savings and reduced environmental impact.
- **Visual Comfort:** Advanced light measurement tools allow for the creation of human-centric lighting designs that consider factors such as color temperature and circadian rhythms. This approach enhances well-being and productivity by mimicking natural light patterns and creating more comfortable and healthier indoor environments.
- **Aesthetic & Functional Design:** The ability to measure and analyse light accurately has allowed architects and designers to integrate lighting seamlessly into buildings and spaces. This has led to innovative lighting solutions that enhance both the aesthetic and functional aspects of environments.

Thus, the evolution of light and measurement technology has profoundly impacted our ability to understand, harness, and optimize light. From ancient firelight to modern LED systems, and from visual estimates to sophisticated spectroradiometers, the advancements in this field have enabled significant progress in science, industry, and everyday life.

CHAPTER 2: BACKGROUND OF THE WORK (LITERATURE REVIEW)

The research and development of Bluetooth-controlled remote cars equipped with advanced sensors has opened new areas in lighting industry, robotics and environmental monitoring. These systems can navigate to various locations, measure illuminance, and capture RGB values, providing valuable data for applications ranging from lighting quality assessment to environmental sensing. By integrating sensors like the BH1750 for illuminance and the TCS3200 for RGB values, these mobile platforms offer a versatile solution for real-time data collection and analysis. This literature review aims to provide a comprehensive overview of the existing research and technologies that underpin the design and functionality of a Bluetooth-controlled remote car with illuminance and RGB measurement capabilities. Specifically, it will explore the evolution of Bluetooth technology, the principles of robotics, the functionality of light sensors, and the methods for processing and interpreting sensor data. By analysing these areas, the review will highlight the current state of the art and identify gaps that the present research seeks to address.

2.1 BLUETOOTH TECHNOLOGY & REMOTE-CONTROL SYSTEMS

Bluetooth technology has undergone significant evolution since its inception, transforming from a simple cable-replacement technology to a robust, low-power wireless communication standard. Haartsen (2000) [14] details the foundational aspects of Bluetooth, including its radio frequency technology, network topology, and application profiles . Kraemer and Bril (2005) [15] provide a comprehensive overview of Bluetooth technology's development, highlighting its adoption in mobile devices and other consumer electronics. Bluetooth's application in remote control systems is extensive due to its reliability, low power consumption, and ease of integration. In mobile robotics, Bluetooth is often used for remote control and data transmission, offering a flexible solution for controlling and monitoring devices over short distances. Research by Kim and Kim (2008) [16] illustrates the use of Bluetooth in the development of a remote-control system for mobile robots, emphasizing its practical advantages. While Bluetooth is well-suited for many applications, it is important to compare it with other wireless technologies such as Wi-Fi, Zigbee, and RF to understand its relative strengths and weaknesses. Bluetooth excels in low-power scenarios and offers sufficient data transfer rates for most remote-control applications, whereas Wi-Fi is better suited for high-bandwidth applications. Zigbee, known for its low power and long battery life, is often used in sensor networks. Such comparisons help in selecting the appropriate technology based on specific application requirements.

2.2 ROBOTICS & REMOTE-CONTROLLED CARS

Mobile robots have undergone significant advancements over the years, fueled by progress in sensor technology, computing power, and control algorithms. Initially, mobile robots were limited to simple tasks in controlled environments. However, with the advent of more sophisticated sensors and algorithms, modern mobile robots are capable of operating autonomously in dynamic and unstructured environments [17] (Siegwart et al., 2011). Mobile robots can be classified into various categories based on their locomotion mechanisms, such as wheeled, tracked, legged, and aerial robots. Each type of robot offers unique advantages and is suitable for specific applications. For example, wheeled robots are well-suited for indoor navigation and logistics tasks, while legged robots excel in traversing rough terrain and navigating obstacles.

Remote-controlled cars serve as a popular platform for experimenting with mobile robotics concepts due to their simplicity, affordability, and versatility. These cars typically consist of a chassis, motor drivers, microcontrollers, and communication modules. The chassis provides the physical structure for mounting components, while the motor drivers control the movement of wheels or tracks. Microcontrollers, such as Arduino or Raspberry Pi, serve as the brain of the system, executing control algorithms and processing sensor data. Communication modules, often based on Bluetooth or Wi-Fi technology, enable remote control and data transmission between the car and the user interface. The design considerations for remote-controlled cars include power management, sensor placement, and data processing. Efficient power management is essential to prolong the car's operating time and ensure uninterrupted operation. Sensor placement plays a crucial role in optimizing sensor coverage and minimizing blind spots. Data processing involves analysing sensor data, executing control algorithms, and making decisions in real-time. The choice of processing hardware and software influences the car's performance and capabilities. Remote-controlled cars are utilized in research for developing and testing algorithms in navigation, sensor fusion, and autonomous control. In industry, they are used for tasks such as warehouse management, inspection, and delivery services. The flexibility and adaptability of these platforms make them valuable tools in both domains.

2.3 LIGHT SENSORS & MEASUREMENT TECHNIQUES

Light measurement is fundamental in various fields such as photography, lighting, and display technology. It involves parameters such as illuminance, luminance, and color temperature. Illuminance refers to the amount of light incident on a surface, while luminance measures the brightness of a light source. Color temperature, measured in Kelvins (K), describes the color characteristics of light emitted by a source, ranging from warm (yellowish) to cool (bluish) light (Wyszecki & Stiles, 2000) [20]

The BH1750 is a digital light sensor commonly used for illuminance measurement. It provides accurate readings in various lighting conditions and offers high resolution and accuracy. The BH1750 sensor is widely used in applications requiring reliable light measurements, such as ambient light monitoring in buildings, streetlights, and electronic devices (Yokogawa Electric Corporation, 2013) [18].

The TCS3200 sensor is capable of detecting RGB values and converting them into frequency output. It consists of photodiodes and filters for red, green, blue, and clear light detection. The TCS3200 sensor is programmable and can be used in various color detection applications, including color sorting, color calibration, and color matching (Texas Advanced Optoelectronic Solutions, 2008) [19].

Light measurement has applications in various industries, including architecture, automotive, and horticulture. In architecture, illuminance measurement is critical for designing well-lit spaces and optimizing energy efficiency. In horticulture, light sensors are used to monitor and control artificial lighting in greenhouse environments, promoting plant growth and productivity.

2.4 COLOR TEMPERATUE & COLORIMETRY

Color temperature (CCT) is an important aspect of lighting quality, describing the color characteristics of light emitted by a source. It is measured in Kelvins (K) and ranges from warm (yellowish) to cool (bluish) light. Wyszecki and Stiles (2000) provide a detailed explanation of color science, including the concept of color temperature and its significance. Calculating CCT from RGB values involves converting the RGB readings to a chromaticity space and then determining the corresponding CCT. Judd and Wyszecki (1975) [21] discuss the methods and mathematical models used in this conversion process . Accurate CCT calculation is essential for applications in lighting design and quality control.

Color measurement has applications in various industries, including manufacturing, printing, and lighting. Accurate color measurement ensures product consistency and quality. In mobile robotics, color sensors like the TCS3200 enable robots to perform tasks such as sorting and quality inspection based on color.

2.5 INTEGRATION OF SENSORS IN MOBILE ROBOTICS

Integrating multiple sensors into a mobile robot requires effective sensor fusion techniques to combine data from different sources. Thrun, Burgard, and Fox (2005) [22] discuss probabilistic robotics, which includes methods for integrating and interpreting sensor data to improve the robot's perception and decision-making abilities. Several case studies highlight the successful integration of sensors in mobile robots. These examples demonstrate how sensor data fusion enhances the robot's functionality and performance in tasks such as navigation, obstacle avoidance, and environmental monitoring. Siciliano and Khatib (2016) [23] provide a comprehensive overview of various sensor integration techniques used in modern robotics.

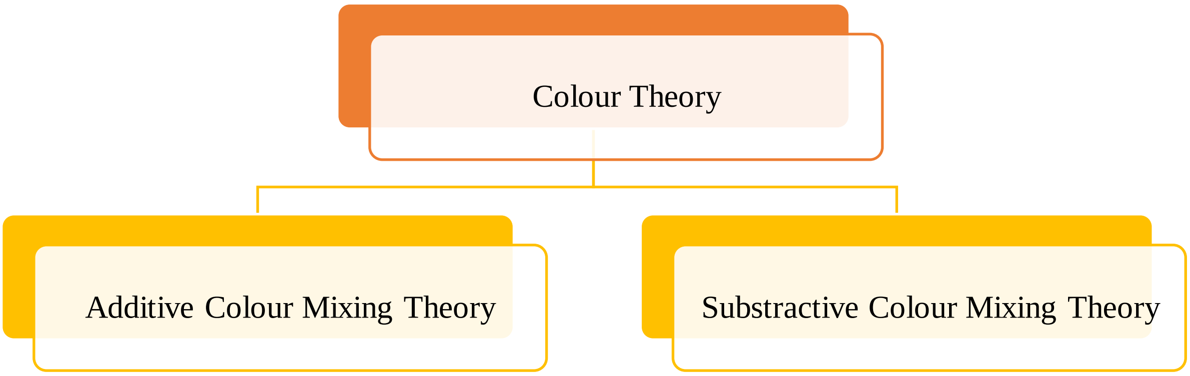
2.6 ALGORITHMS FOR SENSOR DATA PROCESSING

Developing software for sensor data processing involves using programming languages and platforms that support real-time data analysis and control. Bradski and Kaehler (2008) [24] discuss the use of OpenCV, a popular computer vision library, for processing image and sensor data in robotics. Effective data analysis algorithms are essential for interpreting sensor data and making informed decisions. Russell and Norvig (2021) [25] provide an overview of various artificial intelligence algorithms that can be applied to process and analyse sensor data in mobile robotics . These algorithms include filtering, calibration, and machine learning techniques.

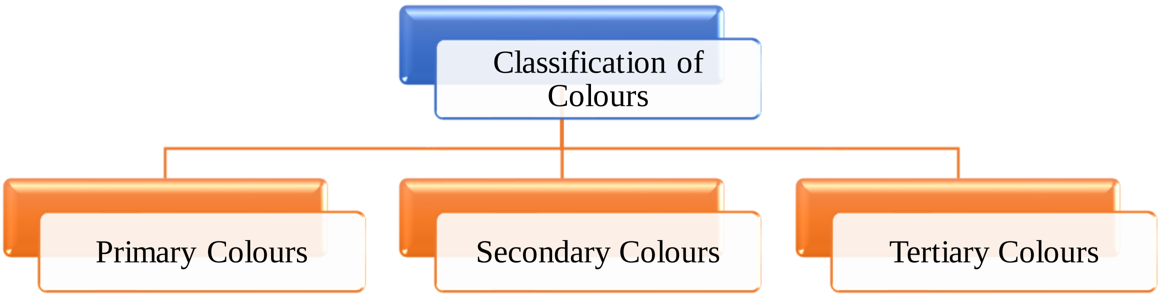
Thus the literature review has provided an overview of the key areas related to the development of a Bluetooth-controlled remote car equipped with illuminance and RGB sensors. By examining the existing research on Bluetooth technology, mobile robotics, light sensors, colorimetry, sensor integration, and data processing, this review has highlighted the current state of the art and identified gaps that the present research seeks to address. This foundation will support the development and implementation of the proposed mobile platform, ensuring it is based on proven technologies and methods.

CHAPTER 3: FUNDAMENTALS OF COLOUR THEORY

Color-Theory is the integration of both science and art of using color. It also explains how humans perceive colour; and the visual effects of how colours mix, match or contrast with each other. Colour theory also involves the messages colours communicate; and the methods used to replicate colour. There are two popular theories which have been given by scientists for understanding the concept of colour mixing [28] those are:



Basically, in colour theory colours are differentiated in three different categories; which are:



3.1 ADDITIVE COLOUR MIXING THEORY

The Additive Colour Mixing Theory [29] , which is often credited to James Clerk Maxwell, is based off the Young-Helmholtz theory of trichromatic vision. The key concept of additive colour mixing theory is that White colour is the mixing of three primary colours (Red, Green, Blue). Additive primary colours are those colours from which all other colours of light are made. One can see evidence of this on their computers, televisions and mobile phones. Pixels emit light in different colours based on how much of each primary colour is emitted from each pixel (RGB values). Here Yellow, Cyan, Magenta are called as the additive secondaries. We can refer to the below figure for better understanding.

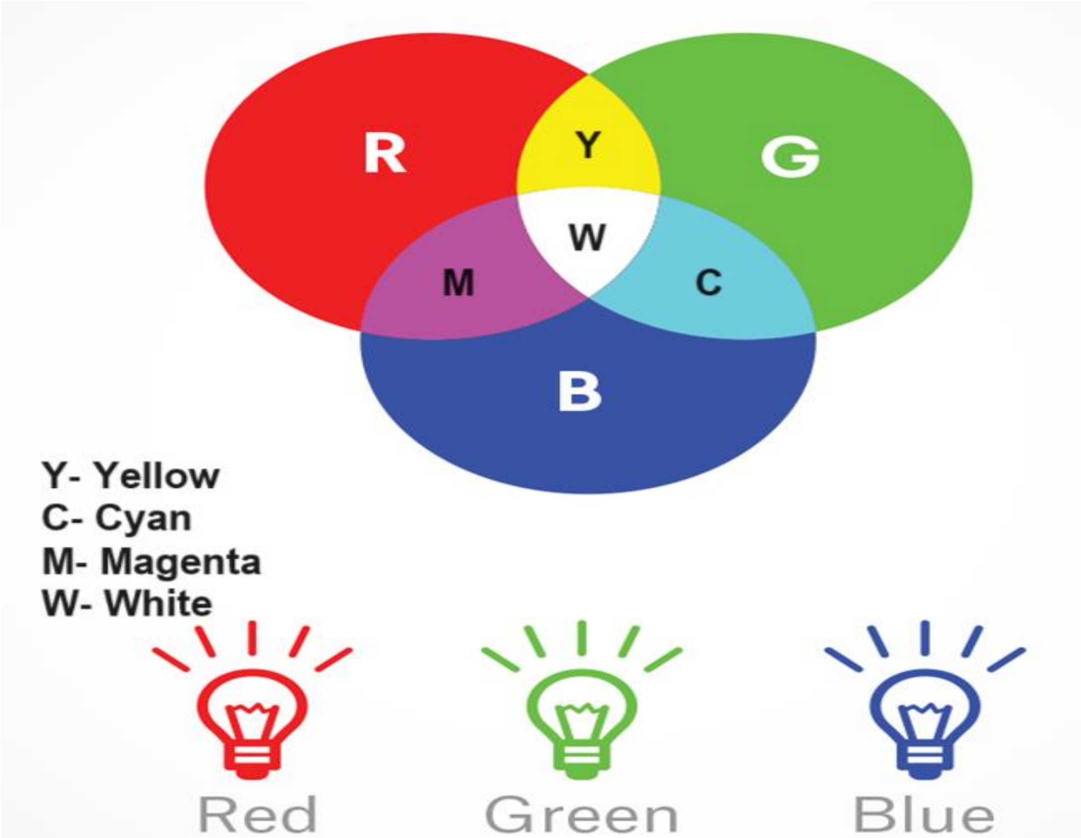


Figure 2: Additive Colour Mixing Theory [30]

3.2 SUBSTRACTIVE COLOUR MIXING THEORY

Substractive Color Mixing Theory [29] was proposed by scientist **Percy Douglas Brewster** in 1913. In this theory Substrative Primaries are Cyan, Magenta, Yellow. And Red, Green, Blue is considered as Substrative Secondaries. The key concept of substractive colour mixing theory is that Black colour is the mixing of three primary colours. This model is also called as CMYK Model.

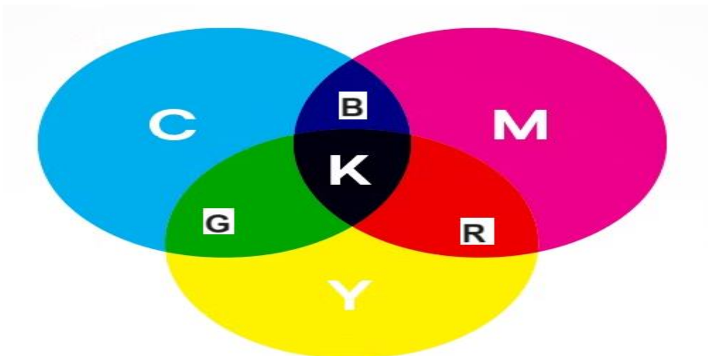


Figure 3: Substractive Colour Mixing Theory [30]

3.3 RGB COLOUR MODEL & COLOUR TUNEABLE LEDs

A colour model is a mathematical representation used to create colours by blending different hues together. In simpler terms, various colours combine to generate a new colour. For example, the mixture of green and blue results in the formation of yellow. This model is highly efficient and swift, attributes often associated with the LED controller.

It's important to note that RGB (Red, Green, Blue) is not the sole colour model available. The reason for this diversity is the remarkable capability of LEDs, which can produce an extensive palette of up to 16 million colours. Through advanced colour-mixing technologies, the primary colours (blue, red, and green) can be combined in various ways to generate a multitude of colours. The LED display's capacity to produce colours is influenced by factors like pixel density and the type of LED used. Consequently, this results in a wide spectrum of colours known as the colour gamut. In essence, the ability of LEDs to create a broad range of colours is a result of intricate colour mixing, with the specific hues produced being influenced by the display's characteristics such as pixel density and the type of LED technology employed.

Figure 4: Colour Tuneable LED

There are three primary types of RGB lights:

1) Classic RGB:

This is the standard RGB configuration, consisting of three LEDs in blue, red, and green. These three primary colors combine to produce a variety of hues.

2) RGB + W:

In addition to the blue, red, and green diodes, RGB + W incorporates a white diode, enhancing the colour spectrum and allowing for the creation of pure white light.

3) RGBW:

RGBW, similar to RGB + W, utilizes four LEDs within the same structure. However, in RGBW, the white diode is integrated rather than being a separate component. This design contributes to the versatility of the lighting system.

➢ **WORKING PRINCIPLE:**

RGB lighting operates on principles akin to those used in forming colours in digital displays. The presence of multiple LEDs allows for the generation of a broad spectrum of colours. An LED controller manages these luminaires, offering presets for colours, intensities, and various settings. These RGB lights feature a diffuser, a component that softens the intensity of the emitted light. Essentially, it prevents the light from being too harsh, resulting in a more subdued and visually appealing illumination.

In terms of efficiency, RGB lighting surpasses incandescent bulbs by over 90%. This remarkable efficiency is primarily attributed to the high lumen efficacy of these luminaires. Lumen efficacy is a metric that gauges how effectively a light source produces visible light in relation to the power consumed.

Furthermore, the efficiency of RGB lights is heightened by a key property of LEDs. LEDs emit light unidirectionally, meaning the light travels in a straight path. This unique characteristic reduces the reliance on reflectors and diffusers to manipulate light. The absence of the need for extensive reflective elements prevents the trapping of light within the luminaire, ensuring the brightness of the emitted light.

3.4 COLOUR SENSOR AND LIGHT MEASUREMENTS

Understanding the above color theories, we can easily relate the operating principle of Color Sensors. A colour sensor is a sophisticated electronic device designed to accurately identify and quantify colours in its surroundings. Functioning on the principles of light absorption, reflection, and transmission, colour sensors play a crucial role in various applications, from industrial automation to consumer electronics. Colour sensors typically consist of multiple photodiodes or photodiode arrays, each sensitive to a specific range of wavelengths in the electromagnetic spectrum. This spectral sensitivity allows the sensor to distinguish between different colours based on the amount of light absorbed or reflected. Many colour sensors use the **RGB model**, with separate channels for red, green, and blue light. By measuring the intensity of light in each channel, the sensor can analyse the colour composition of the incident light. This is especially common in consumer electronics and displays. Colour sensors often employ sophisticated algorithms to interpret the data collected from different spectral channels. These algorithms help in accurate colour reproduction by comparing the sensor's readings to predefined colour profiles. The Colour Sensor we have used in this project is **TCS3200** [31] so let's discuss briefly about that colour sensor.

3.4.1 FUNDAMENTALS OF TCS3200 SENSOR

The TCS3200 is an advanced device designed as a programmable color sensor. It functions as a light-to-frequency converter, bringing together customizable silicon photodiodes and a current-to-frequency converter on a single integrated circuit using CMOS technology.

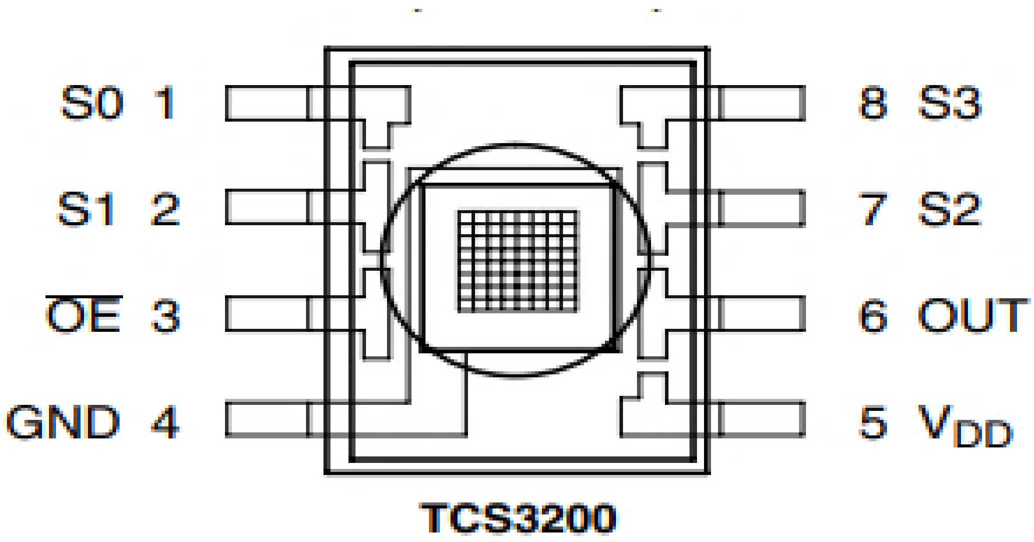


Figure 5: Diagram of TCS3200 Colour Sensor [31]

This integration enables a compact and efficient solution for capturing and interpreting color information. In simpler terms, the TCS3200 takes incoming light, utilizes silicon photodiodes to sense different colors, and then converts this information into a square wave output. The frequency of this square wave is directly linked to the intensity of the detected light, or in more technical terms, the irradiance. The output is a square wave with a 50% duty cycle, meaning an equal duration of high and low states in each cycle.

The TCS3200 is equipped with the capability to adjust its full-scale output frequency using three preset values, which can be selected through two control input pins. This feature provides flexibility in tailoring the output to specific requirements. The device facilitates easy interfacing with a microcontroller or other logic circuitry through its digital inputs and outputs.

Moreover, the Output Enable (OE) function allows the user to place the output in a high-impedance state. This is particularly useful when multiple TCS3200 units share a microcontroller input line, enhancing efficiency and ease of integration. In terms of its internal structure, the TCS3200 incorporates an 8 × 8 array of photodiodes within its light-to-frequency converter. Among these photodiodes, 16 are equipped with blue filters, another 16 with green filters, 16 with red filters, and the remaining 16 are clear, lacking any filters. This arrangement allows the sensor to discern and quantify light across different color spectra, providing a comprehensive and versatile solution for color detection and analysis.

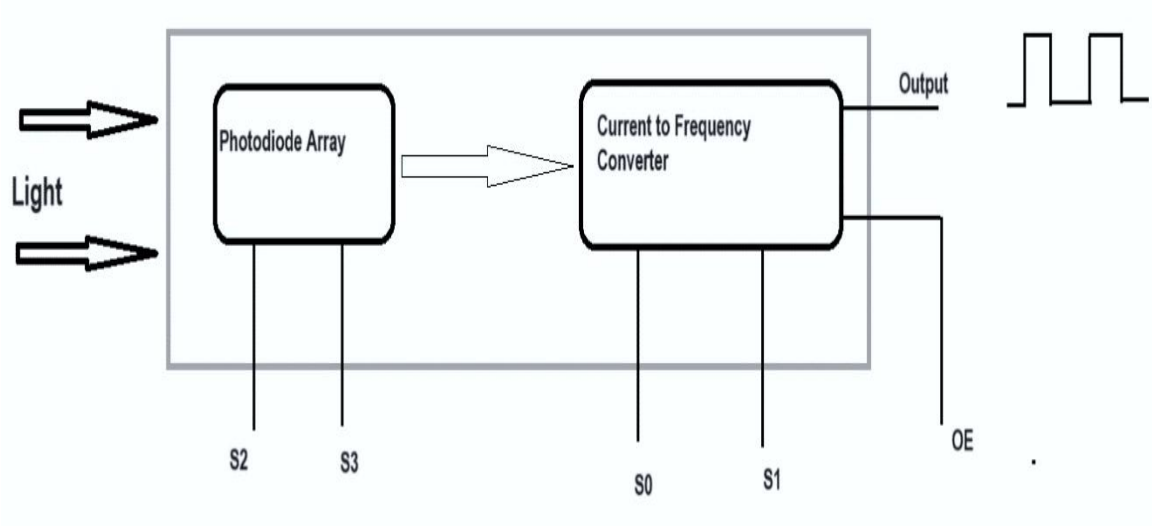


Figure 6: Functional Block Diagram of TCS3200 Sensor [31]

➢ **Selectable Options for Frequency Scaling:**

For frequency scaling there are few scalable options available as per the High (H) level signal and Low (L) level signal in the terminals S0, S1 respectively.

S0	S1	OUTPUT FREQUENCY SCALING (f_0)
L	L	Power Down
L	H	2 %
H	L	20 %
H	H	100 %

S2	S3	PHOTODIODE TYPE
L	L	RED
L	H	BLUE
H	L	CLEAR
H	H	GREEN

3.4.2 CHARACTERISTIC RESPONSE OF TCS3200:

The Characteristic graph of photodiode spectral responsivity is as shown below;

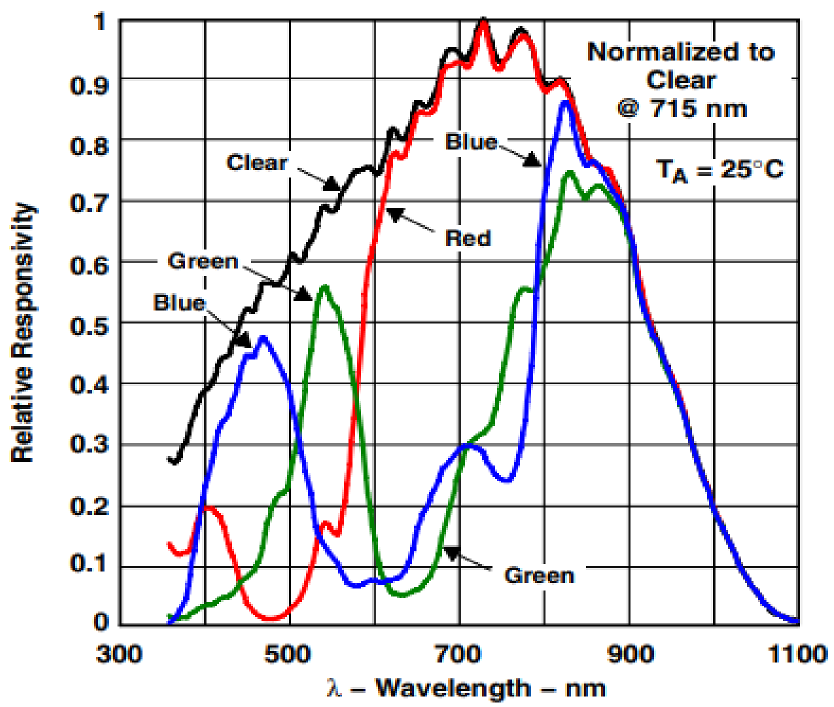


Figure 7: Relative Responsivity Vs. Wavelength for TCS3200 [31]

CHAPTER 4: FUNDAMENTALS OF PHOTOMETRIC MEASUREMENTS

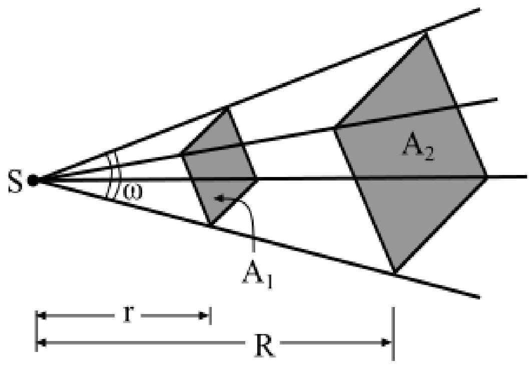
Photometric measurements play a vital role in ensuring the quality and development of lighting systems intended for human environments. This encompasses a wide range of applications, from displays in automobiles and televisions to the general lighting used in buildings, including LEDs. Photometry, as previously explained, involves calculating visible light based on how the human eye typically responds. The eye's response is influenced by physical, physiological, and psychological factors, which vary significantly among individuals. Additionally, the human eye possesses two types of cells, rods for low light levels and cones for high light levels, making it challenging to define an average observer. In 1924, The International Commission on Illumination (CIE) [35] conducted experiments to measure the human eye's response to visible light. Through these experiments, they identified two vision response curves crucial for photometric measurements. For high light conditions, the photopic function was established, represented by the spectral luminous efficiency function $V(\lambda)$, characterizing the daylight vision of an average human observer. For low light levels, the CIE introduced the scotopic function $V'(\lambda)$ to represent the dark-adapted eye response. These standardized functions provide a foundation for quantifying and understanding how the human eye perceives light under different conditions [34].

4.1 LAWS OF PHOTOMETRIC MEASUREMENT

There are two principal laws utilized to produce photometric measurements. They are known as :

4.1.1 INVERSE SQUARE LAW

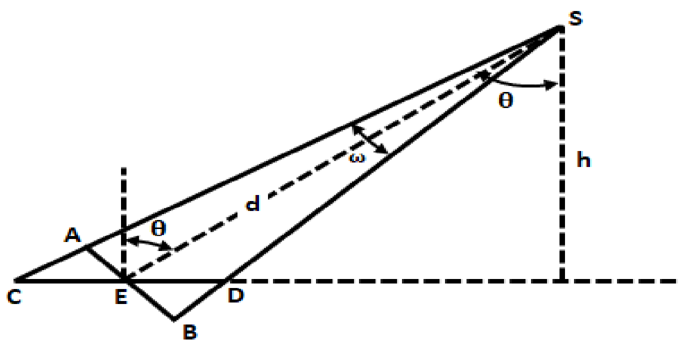
The inverse square law [31] establishes a relationship between the brightness of illumination from a source with constant intensity and its distance from a surface. According to this law, the intensity per unit area on the surface changes in inverse proportion to the square of the distance between the source and the surface. In simpler terms, if the distance from the light source to the surface is doubled, the illumination intensity per unit area becomes one-fourth, and if the distance is halved, the intensity per unit area becomes four times greater. This principal highlight that the precision of consecutive illuminance measurements is directly dependent on how well the distance between the surface and the light source is controlled.



In the scenario depicted in the illustration, we have a point light source labelled 'S' with a luminous intensity of 'I' lumens per steradian. There are two surfaces with areas A1 and A2, placed at distances 'r' and 'R' meters away from the point source, respectively. Both surfaces are enclosed within the same solid angle denoted by 'ω'. Now from the theory we can obtain the expression of illuminance as : $\boldsymbol{E_1 = \frac{I}{r^2}}$; $\boldsymbol{E_2 = \frac{I}{R^2}}$. From this expression, it can be seen that the illumination varies inversely as square of the distance between the source and the surface.

4.1.2 LAMBERT'S COSINE LAW

Lambert's cosine law [32] describes how the brightness of light on a surface with a constant area change with the incident angle. When light strikes a surface at an angle, the illumination of the surface is directly related to the cosine of the angle θ formed between the direction of the incident light and the normal to the surface. In simpler terms, as the angle of incidence increases, the illumination on the surface decreases. A surface following this law maintains consistent radiance when viewed from any angle, displaying what is known as Lambertian reflectance. This means that the brightness perceived by the human eye remains the same, irrespective of the viewing angle. In essence, Lambert's cosine law provides insights into how light interacts with surfaces, influencing their brightness in a predictable manner based on the incident angle.



Lambert's Cosine Law

Considering the above figure we have a point light source 'S' positioned at a height 'h' above the surface CD. We assume that surface CD appears to cover a certain solid angle 'ω' when viewed from the point source 'S'. Additionally, we introduce surface AB, which is inclined and has its centre point marked as E. This surface AB is perpendicular to the direction of light flux. So the illuminance on the surface CD at point E is, $\boldsymbol{E_E = \frac{I}{h^2} \cos^3\theta}$; where $\boldsymbol{\frac{I}{h^2}}$; is the illuminance at Nadir point from the source light.

4.2 PHOTOMETRIC QUANTITIES

Some of the commonly known photometric quantities are Luminous Energy, Luminous Flux, Luminous Efficacy, Illuminance, Luminous Intensity, Luminance and CCT. In the field of photometric measurement these quantities play significant role. In our developed prototype model, we have introduced the feature to measure the CCT and Illuminance level of any light source. Thus, here we will discuss about these two photometric quantities in detail.

4.2.1 ILLUMINANCE (E_v)

- **Definition:**

In photometry, Illuminance is defined as the total incident luminous flux per unit area of a surface. Illuminance indicates the amount of light level that illuminates a particular working plane. E_v is the symbol commonly used to define illuminance level.

- **Unit:**

The SI unit of illuminance is Lux which is also termed as lumens per square meter. Mathematically illuminance is expressed as : $\boldsymbol{E_v = \frac{d\phi}{dA}}$; Where, $d\varphi$ is the differential luminous flux and dA is the differential projected area. Thus, illuminance on a particular working plane is also defined as the ratio between $d\varphi$ and dA.

- **Measuring Device:**

A light meter is a device used to measure the amount of light level in a particular working plane. Photo-electric type light meters are most commonly used for measuring illuminance. BH1750 is one of the most widely used Light Sensor in light measuring devices.

Figure 8 Lux Meter: Illuminance Measuring Device [36]

4.2.2 CORRELATED COLOUR TEMPERATURE (CCT)

- **Definition:** Correlated Color Temperature is being defined as the absolute temperature of a blackbody whose color appearance most nearly resembles that of the light source. In lower temperature the color appears of warm tone which actually ranges between 2700K to 3500K. Whereas the temperature rises the color turns into white and blue spectrum. We see this types in lamp of 4000K to 5000K.
- **Measuring Device:**

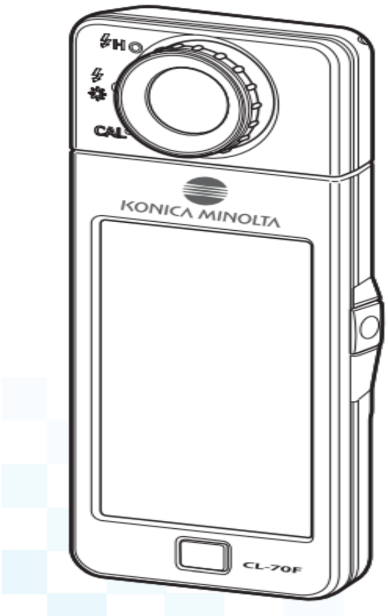
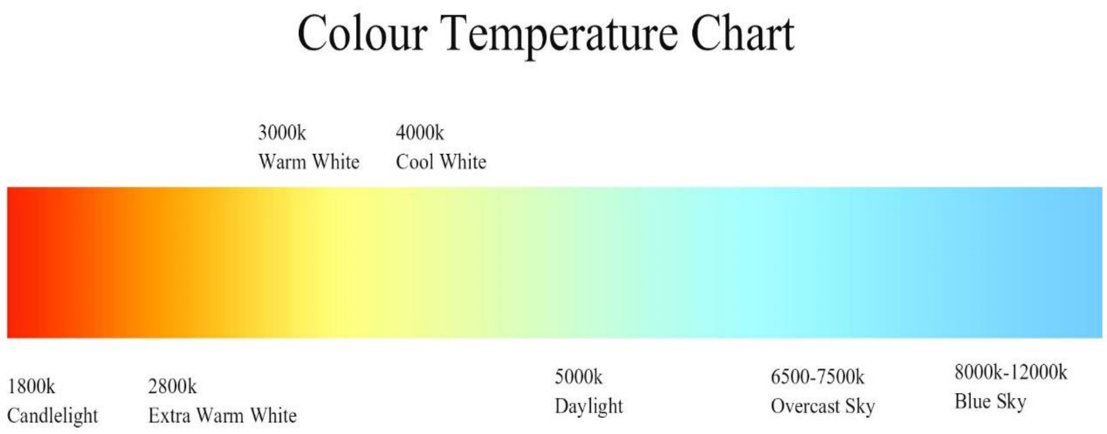


Figure 9: Konica Minolta [37]

➢ Specifying Color Temperature:

Some of the most commonly known color temperature along with their Kelvin rating is given below:



➢ Calculation of CCT:

In CIE 1931 Chromaticity System which is also known as CIE 1931 X-Y-Z Model, where Red Green and Blue are the primary colors that are represented by X,Y and Z respectively. Using this concept McCamy has proposed the formula for CCT calculation. McCamy's model is valid for range of CCTs from 2200K to 10000K. The formula for CCT computation is as given below:

$$T = -437n^3 + 3601n^2 - 6861n + 5524.31$$

Where, $n = \frac{(x-0.332)}{(y-0.1858)}$; which actually derived from the Chromaticity (x, y) Co-ordinates.

But there are certain limitations in applying this formula in calculating the CCT value from the obtained R-G-B responses that we collect from the sensors. The obtained values of R-G-B quantities in any test light source is the frequency response of that particular primary color captured by the RGB Sensor. Thus if we follow McCamy's Model it can provide errors in calculation. Thus we achieve more precise calculation approach we need to refer to other methodologies to derive the Colour Temperature values.

CHAPTER 5: THEORETICAL COMPUTATION OF CCT FROM RESPONSES OF R-G-B SENSOR

Referring to the book "Lighting Engineering: Applied Calculations" written by R. H. Simons and A. R. Bean [38] , we can understand the concept of CCT computation from frequency responses of Red, Green and Blue light content from RGB Sensor. To understand this concept, we will take an example as discussed below.

- **Problem Statement:** Let us consider the measured value of R-G-B responses from the RGB sensor is R = 44, G = 53, B = 64 Now we have to determine the CCT (Correlated Color Temperature) Value.
- **Calculation:** Let us consider CCT is represented as T in this context so the basic formula to calculate CCT from RGB responses is as given below:

$$T(x) = \frac{T_1(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + \frac{T_2(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + \frac{T_3(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} \quad \ldots [1]$$

Now for **Red Response**: Firstly, the given value of R is $x = R_g = 44$ *Here the considered values represents the nearest values of the given R values*

$$\begin{cases} x_1 = R_m = 44 \\ x_2 = R_{m+1} = 46 \\ x_3 = R_{m+2} = 43 \end{cases}$$

Now the values of K_1, K_2 and K_3 can be written as;

$$K_1 = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} = \frac{(44-46)(44-43)}{(44-46)(44-43)} = 1$$

$$K_2 = \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} = \frac{(44-44)(44-43)}{(46-44)(46-43)} = 0$$

$$K_3 = 1 - K_1 - K_2 = 0$$

Similarly for **Green Response**: Firstly, the given value of G is $x = G_g = 53$ *Here the considered values represents the nearest values of the given G values*

$$\begin{cases} x_1 = G_m = 57 \\ x_2 = G_{m+1} = 56 \\ x_3 = G_{m+2} = 54 \end{cases}$$

Now the values of K_1, K_2 and K_3 from the formula can be obtained as 1, -2, 2 respectively.

Similarly for **Blue Responses**: Firstly, the given value of G is $x = B_g = 64$ $Here\ the\ considered\ values\ represents\ \ the\ nearest\ values\ of\ the\ given\ B\ values$

$$\begin{cases} x_1 = \ B_m = 67 \\ x_2 = \ B_{m+1} = 66 \\ x_3 = \ B_{m+2} = 65 \end{cases}$$

Now the values of K_1, K_2 and K_3 from the formula can be obtained as 1, -3, 3 respectively.

Now to determine the CCT the methodology says that between **R**, **G** and **B** values we have to make constant any value (for example let's say **B**) and then we have to vary the other two terms as per necessity in the CCT function. So let us consider **B** as constant term and let's determine the CCT. Considering equation [1] we see;

$$T_1 = CCT\ of\ (R_m, G_m, B_m)$$

$$T_2 = CCT\ of\ (R_{m+1}, G_{m+1}, B_m)$$

$$T_3 = CCT\ of\ (R_{m+2}, G_{m+2}, B_m)$$

Here one thing need to be looked into that to determine the values of T_1 , T_2 , T_3 we have to look into the look-up table that was prepared during RGB sensor calibration. Using that particular look-up table, we have to determine the values of respective CCTs.

Now by considering the look-up table that was given in the sensor calibration part of this report we can now say that: $CCT\ of\ (44, 57, 67) = 2700\ K$

$$CCT\ of\ (46, 56, 67) = 3000\ K$$

$$CCT\ of\ (43, 54, 67) = 3200\ K$$

Now putting the values in equation [1] we can determine the final CCT value as:

$$T = T_1K_1 + T_2K_2 + \ T_3K_3$$

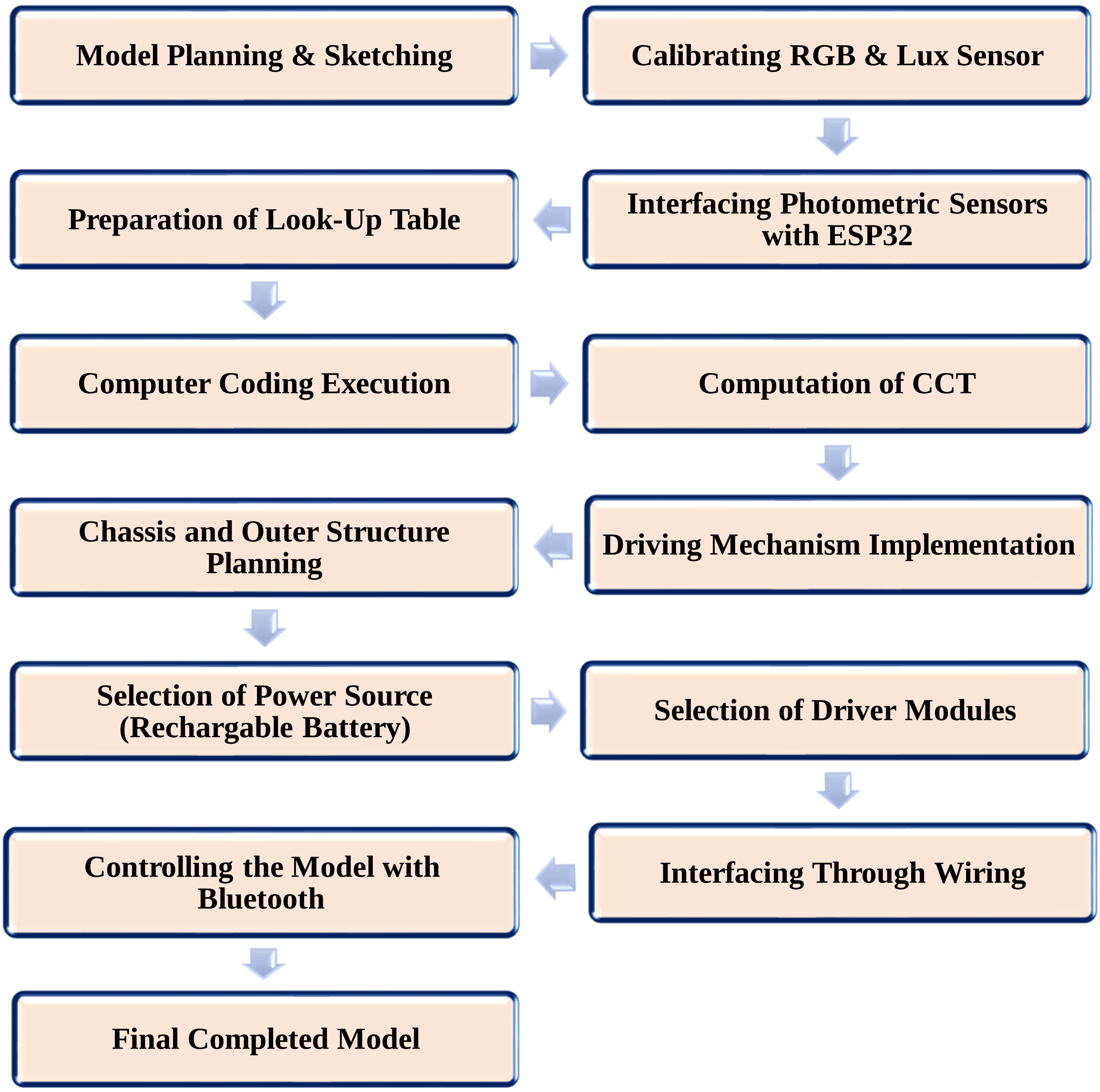
$$T = 1 \times 2700 + (-2) \times 3000 + \ 2 \times 3200$$

So, Correlated Color Temperature, $T = 3100\ K$

Now using this same theoretical concept, we will design an algorithm in computer programming to calculate the CCT of any particular light source using RGB Sensor which has been introduced in Chapter 9 of this paper.

CHAPTER 6: MAKING PROCEDURE OF THE HARDWARE MODEL

Some of the major prerequisites before starting the construction of the project will be Power Electronics, Electric Circuit, Motors, Electrical and Electronic equipment, Drivers, Sensors and of course Robotics. One should have knowledge on these following subjects to implement the project model. Here, in this section we will try to demonstrate the simplest and efficient way of building the hardware construction. So that anyone intending to develop this model or want to give upgradation to this model; would be benefited from it. Here, for better understanding the Making Procedure is divided into few parts according to its features as given below:



CHAPTER 7: CALIBRATION OF RGB SENSOR AND LUX SENSOR

Considering the above components we can construct the sensors set-up for measurement purpose. But to obtain accurate measurements we first need to calibrate our Test Sensors under standard conditions and then we have to verify the test results with the official “Data Sheet” given by the manufacturer to understand whether the sensors are working fine or not. Thus, we can also define the precision of the sensors. The detailed experimental set-up to calibrate the sensors are as discussed below:

7.1 EXPERIMENTAL SET-UP

In this experiment of calibrating the RGB Sensor and Lux Sensor we have prepared a testing environment using a Philips Wiz Wi-Fi 10-Watt LED Smart Bulb and a mini darkroom set-up using a box of black inner surface. Let’s now understand the method and required equipment for this experimental set-up.

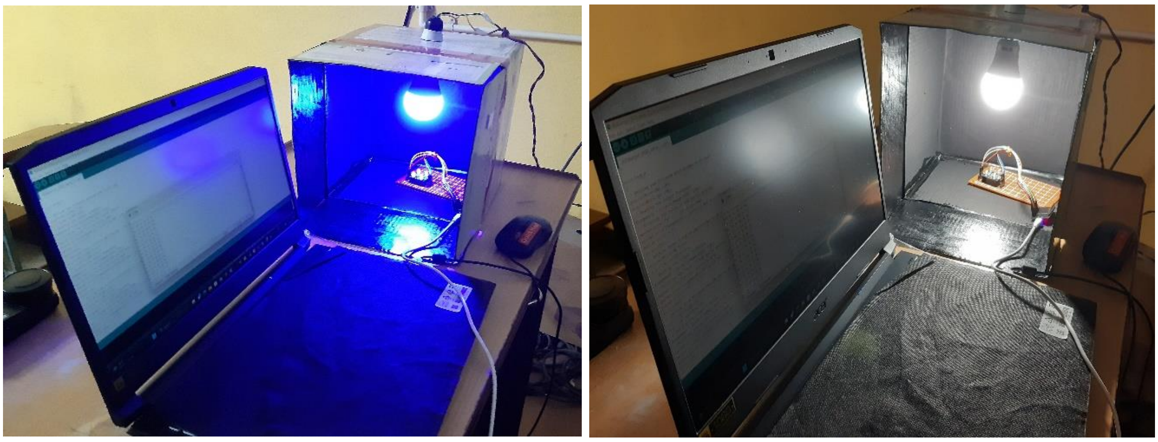


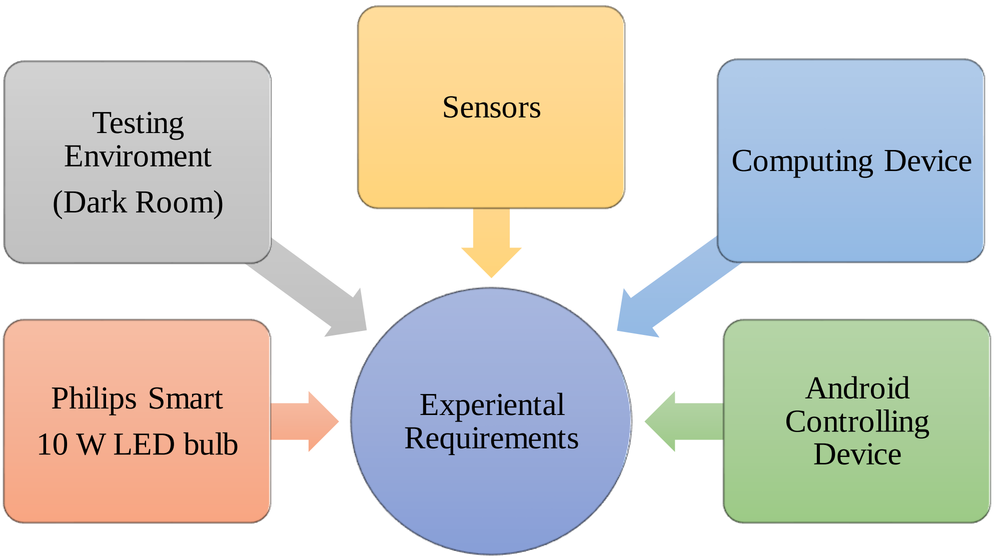
Figure 10: Overview of The Experimental Set-Up

- **Methodology:** The procedure of this experiment is as listed below:

1. Firstly, we should interface the Sensors with ESP32 Module.
2. Then we will have to perform our experiment in the Mini Dark Room.
3. This room has been constructed using a cubic box whose inner surfaces has been colored black to avoid unwanted reflection.
4. Now the holder is placed at the top of the box and the LED smart bulb is fixed.
5. Then we have to mark the “Nadir Point” and have to place the sensors at that point.
6. In this experiment considering all factors to get most accurate measurement we have to set brightness of the light at 20%

7. Then we have to gradually vary the intensity of light and the type of color such as White, Red, Green, Blue in this case and write down the necessary readings.
8. At the same time, we have to verify the same readings using any standard equipment; such as KONICA MINOLTA which is used in this case.
9. For better result we can note down the responses of oscilloscope for each frequency sampling.
10. Finally, we have to tally those results with the sensors datasheet and thus the sensors will be calibrated.
11. Lastly, using those final readings we will prepare the final Look-Up Table which will be used for calculation of CCT.

➢ **Experimental Requirements:**



7.2 EXPERIMENTAL RESULT

Color Type	**CCT** (K)	**Frequency Sampling**	**Intensity**	**R** (Red)	**G** (Green)	**B** (Blue)	**Illuminance** (Lux)
White Light	5500	2 %	100 %	1925	2345	2421	307
			80 %	2080	2640	2526	320
			60 %	2696	3115	3142	330
			40 %	3463	4118	4575	165
		20 %	100 %	39	47	48	237
			80 %	35	42	43	230
			60 %	32	39	40	181
			40 %	31	40	41	125

		100 %	100 %	39	49	50	328
			80 %	35	43	44	323
			60 %	34	43	44	250
			40 %	33	42	43	158

Color Type	**CCT** (K)	**Frequency Sampling**	**Intensity**	**R** (Red)	**G** (Green)	**B** (Blue)	**Illuminance** (Lux)
Red Light	1000	2 %	100 %	6766	14189	11884	7.8
			80 %	7560	14282	12114	6.4
			60 %	5342	14402	12350	5.3
			40 %	12142	14691	12950	6.9
		20 %	100 %	143	290	242	43.9
			80 %	138	288	241	10.5
			60 %	248	297	262	5.5
			40 %	133	288	255	5.1
		100 %	100 %	138	285	338	8.3
			80 %	161	284	237	8.2
			60 %	172	281	235	7
			40 %	132	282	250	6.5

Color Type	**CCT** (K)	**Frequency Sampling**	**Intensity**	**R** (Red)	**G** (Green)	**B** (Blue)	**Illuminance** (Lux)
Green Light	7000	2 %	100 %	8698	10423	11076	13.3
			80 %	9436	11364	11521	12.4
			60 %	9955	12244	11970	8.6
			40 %	10632	13313	12384	5.8
		20 %	100 %	232	185	216	18.9
			80 %	233	184	216	16.1
			60 %	235	187	234	9.8
			40 %	247	298	261	7.2
		100 %	100 %	146	186	199	16.6
			80 %	153	189	257	13.8

			60 %	240	279	255	7.1
			40 %	241	290	257	6.6

Color Type	**CCT** (K)	**Frequency Sampling**	**Intensity**	**R** (Red)	**G** (Green)	**B** (Blue)	**Illuminance** (Lux)
Blue Light	5500	2 %	100 %	12888	11802	7544	2.1
			80 %	11935	12412	8397	1.4
			60 %	12224	14682	12907	1.3
			40 %	12230	14690	12907	1.1
		20 %	100 %	230	286	254	3.2
			80 %	244	293	257	2.3
			60 %	246	294	263	2.1
			40 %	246	291	256	1.6
		100 %	100 %	233	238	252	2.1
			80 %	233	239	227	1.5
			60 %	244	294	229	1.2
			40 %	246	295	258	1

7.3 EXPERIMENTAL RESULT ANALYSIS

In this section, we analyse the results obtained from the experiment designed to calibrate the RGB and Lux sensors using a controlled environment. This analysis aims to understand the sensors' response to different lighting conditions and verify the accuracy of the readings against standard equipment. Based on the different color type of the light source we have analysed the above readings, which has been shown below:

- **White Light :**

1. The Lux values increase as the intensity increases, showing a direct correlation.
2. The RGB values slightly increase with higher intensity, reflecting the sensor's ability to capture subtle changes in light.

- **Red Light:**

1. The Lux values are lower compared to white light, indicating lower light intensity.
2. Significant variations in RGB values with intensity changes highlight the sensor's sensitivity to red light.

- **Green Light:**

1. Green light results in higher Lux values compared to red light, reflecting its higher intensity.
2. The RGB values vary consistently with intensity changes, showing sensitivity to green light.

- **Blue Light:**

1. Blue light results in the lowest Lux values, indicating the lowest intensity.
2. RGB values show variations similar to other colors, confirming the sensor's sensitivity to changes in blue light intensity.

- **Overall Analysis:**

1. **Lux Values:** The Lux Values are consistently increase with increasing light intensity for all color types. It is higher for white light, followed by green, red, and lowest for blue light.
2. **R-G-B Values:** It shows significant variation with changes in light intensity and color types. Moreover, highlight the sensor's sensitivity and ability to distinguish between different light conditions and colors.

The experimental results provide detailed insights into the performance of the RGB and Lux sensors. The sensors exhibit sensitivity to variations in light conditions and colors, validated through consistent changes in Lux and RGB values with light intensity. The data demonstrates the effectiveness of the calibration process. It also ensures accurate measurements across different colors and intensities, which is crucial for applications like CCT calculation.

CHAPTER 8: SYSTEM DEVELOPMENT PART I : RGB RESPONSE & ILLUMINANCE MEASUREMENT

In this section of the paper, we will see a detailed analysis of the prototype model based on its key features. Here we will understand about different components used in model making, their specifications, experimentation, data tables, results and many more. Let's now discuss about system development in details.

8.1 SYSTEM OVERVIEW

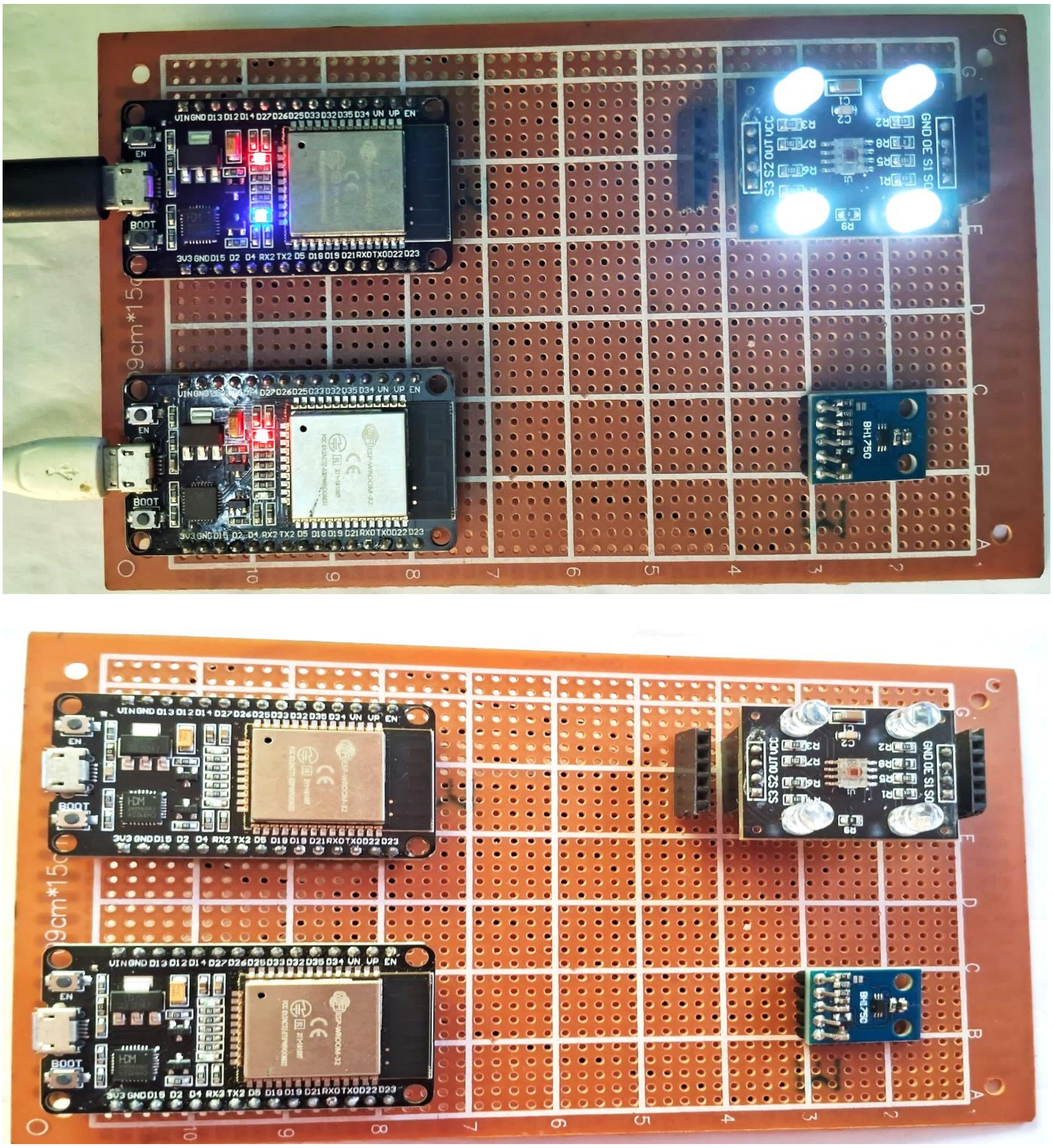


Figure 11: Sensors(TCS3200 & BH1750) Setup Overview

8.2 REPRESENTATION OF CIRCUIT CONNECTION

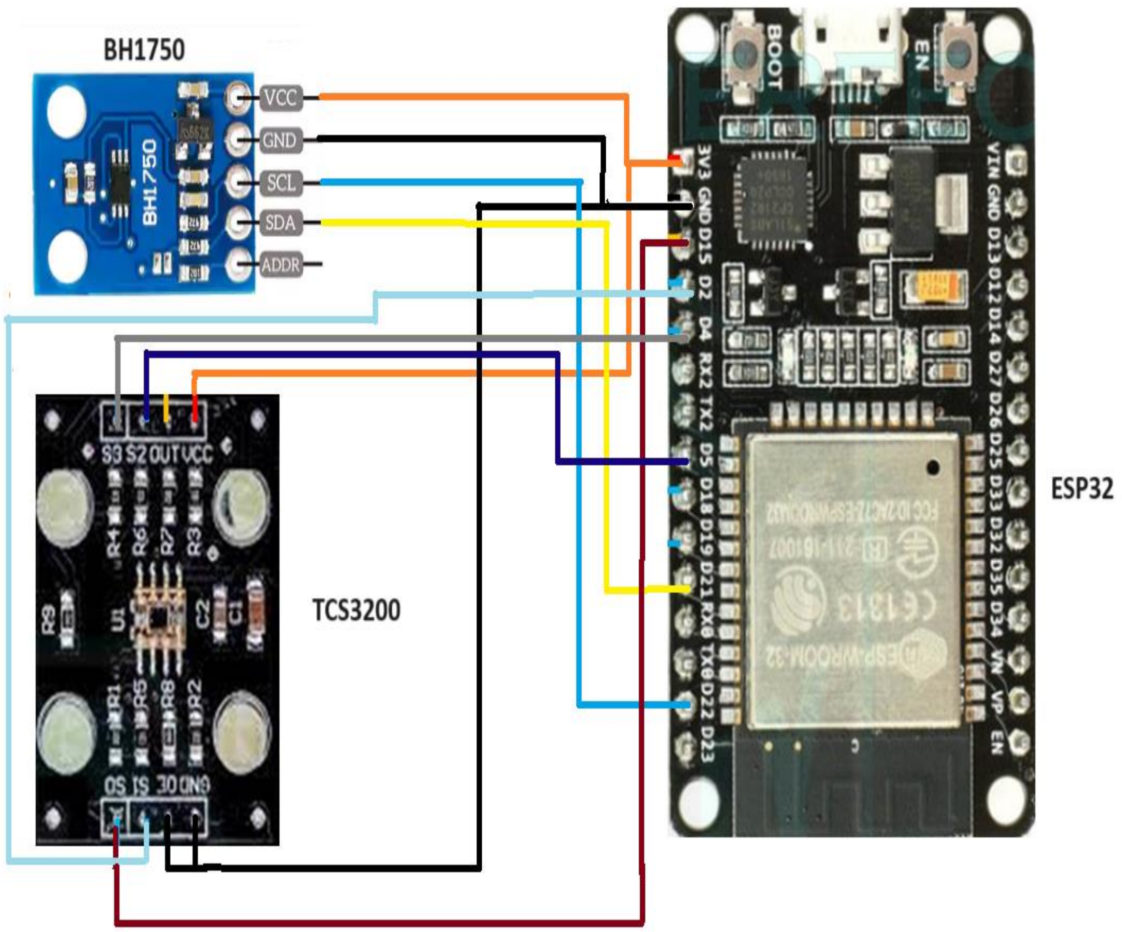
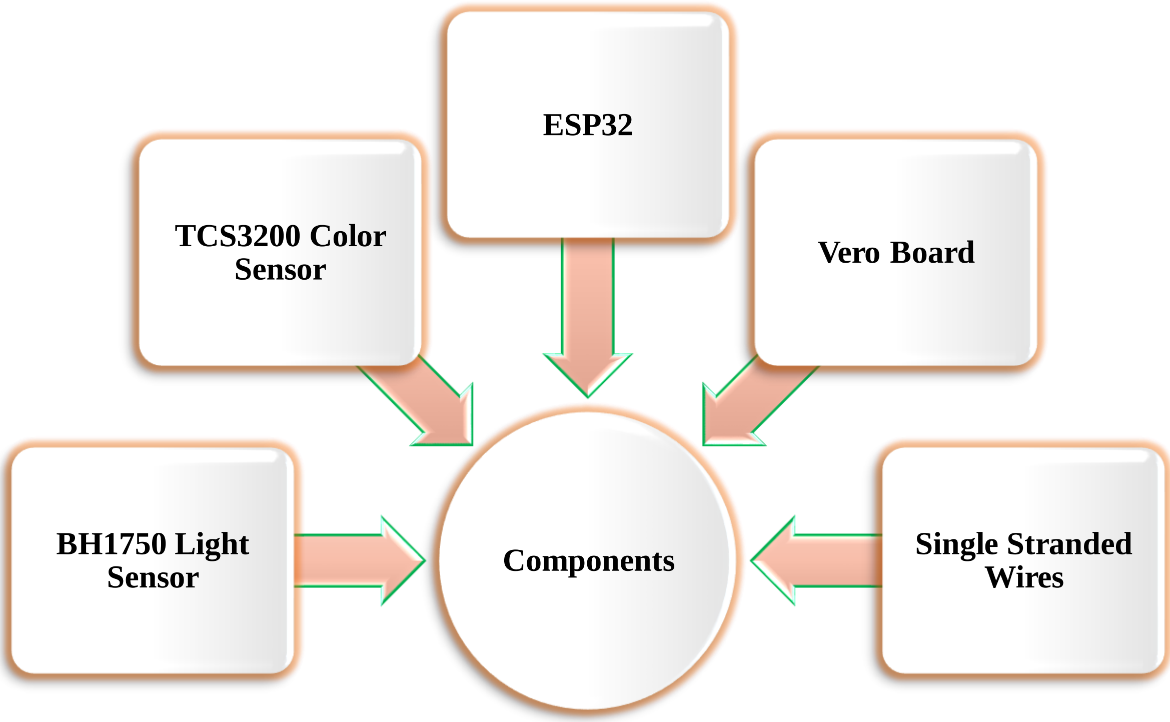


Figure 12: Integrated Circuit Connection

8.3 COMPONENTS WITH SPECIFICATIONS



we can see in the “Model Overview” that several components have been used in making of the model. The components has significant role in completion of the model. The specification of each component along with their key features are listed below. The flow chart shows each component for this particular section of discussion. Let’s now discuss about all these components in details.

8.3.1 BH1750 LIGHT SENSOR:

- **Description:**

The BH1750 Module is a digital Ambient Light Sensor integrated circuit designed specifically for use with the I2C bus interface. It is particularly well-suited for obtaining ambient light data to adjust the power of LCD and keypad backlights in mobile phones. With its high-resolution capabilities, it can accurately detect a wide range of light levels, from 1 to 65535 lux. This makes it an ideal choice for applications where precise ambient light measurement is essential for optimizing display visibility and power consumption in mobile devices.

- **Specifications:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
BH1750	Length (mm)	18	1
	Width (mm)	13	
	Height (mm)	3	
	Weight (g)	5	
	Operating voltage (VCC)	3 V to 5V	
	Operating current	120 µA to 190 µA (max)	
	Operating Temperature	-40°C ~ 85°C	
	Sensor Resolution	16 Bits	
	Peak wavelength	560 nm	
	Interface Type	12C	
	I2C Maximum Clock frequency	400 KHz	

- **Component Diagram:**

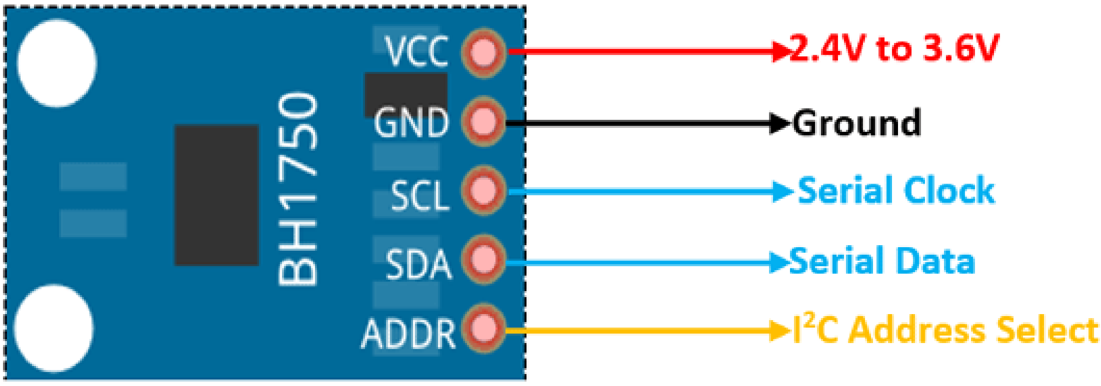


Figure 13: BH1750 Ambient Light Sensor Overview [48]

- **Features:**

1. I2C bus Interface (f / s Mode Support)
2. Spectral responsibility is approximately human eye response
3. Illuminance to Digital Converter
4. Wide range and High resolution. (1 - 65535 lx)
5. Low Current by power down function and 50Hz / 60Hz Light noise reject-function
6. 1.8V Logic input interface and Light source dependency is little. (ex. Incandescent Lamp. Fluorescent Lamp. Halogen Lamp. White LED. Sun Light)
7. It is possible to select 2 types of I2C slave-address.
8. Adjustable measurement result for influence of optical window.
9. Small measurement variation (+/- 20%) and the influence of infrared is very small.

- **Applications:**

1. Widely used to measure Lux level in any interior/exterior space in Lighting Design.
2. Manipulate brightness in mobile phones/LCD
3. Automobile Headlights on/off control based on surrounding lighting conditions.
4. Used as Ambient light sensors to control the brightness of Display screens.

- **Working Principle:**

The hardware setup of the BH1750 module is straightforward, centered around the BH1750FVI IC as the main component. Operating at a voltage of 3.3V, a voltage regulator is incorporated to ensure proper power supply. Additionally, pull-up resistors of 4.7K are utilized for the I2C lines. To obtain lux values from the BH1750, communication occurs via the I2C bus. Within the IC, an analog-to-digital converter (ADC) transforms the analog illuminance into digital lux values. These lux values are then transmitted to the microcontroller via the I2C pins, with the SCL pin providing clock pulses and the SDA pin facilitating data transfer. The BH1750 IC employs a photodiode that mimics the response of the human eye, ensuring accurate light detection. Furthermore, an internal oscillator within the IC regulates the clock signal for the internal logic of the IC. This setup enables seamless communication between the BH1750 sensor and the microcontroller, facilitating precise ambient light measurement and adjustment in various applications.

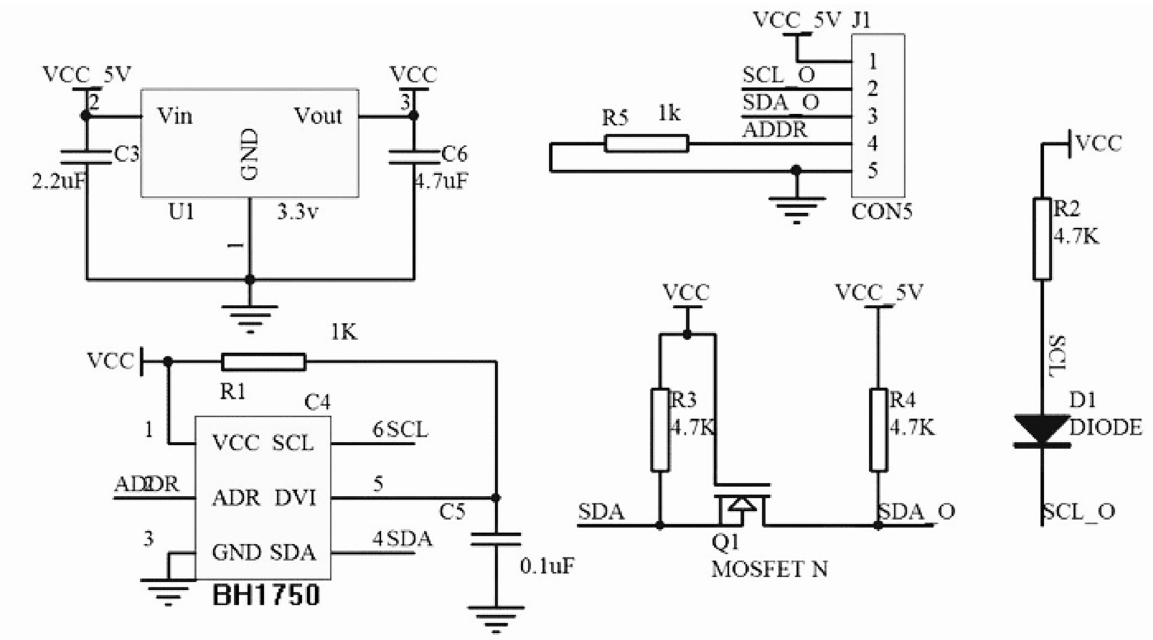


Figure 14: Functional Diagram [49]

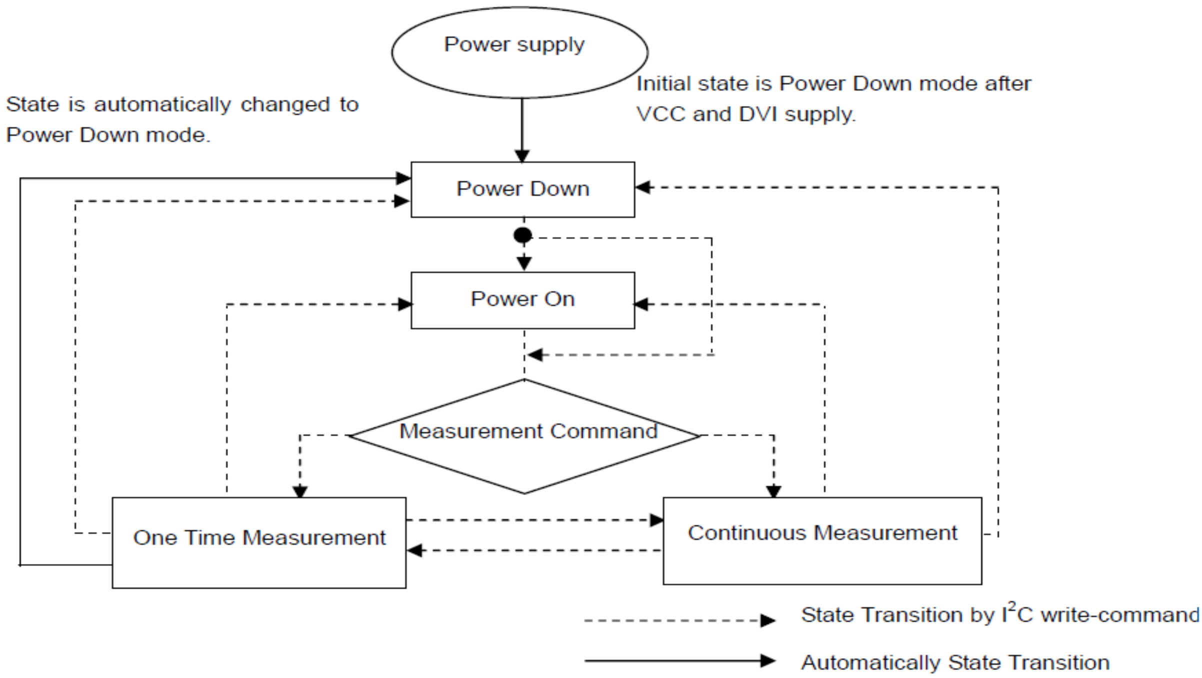


Figure 15: Measurement Procedure of BH1750 [49]

➢ Interfacing BH1750 With ESP32:

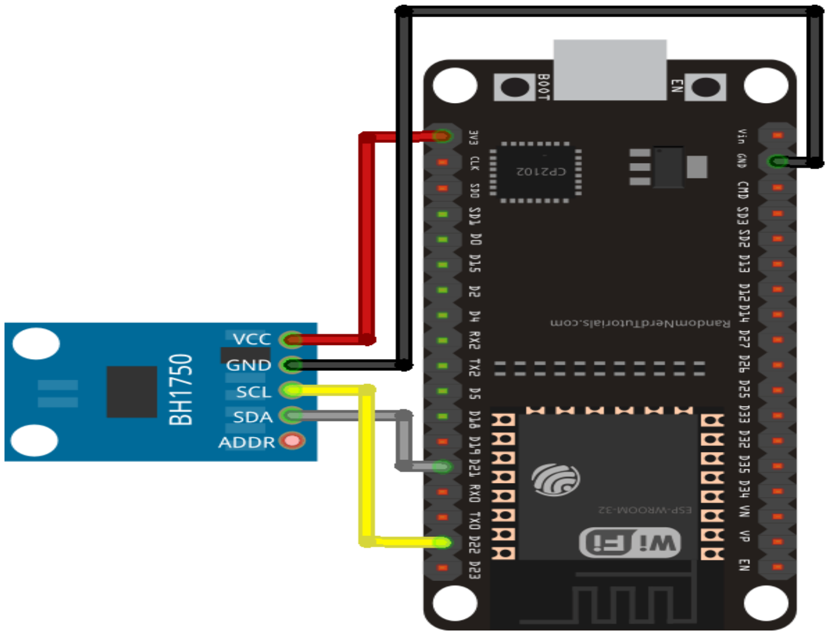


Figure 16: Connection with ESP32

➢ Uploading Code (Arduino IDE):

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <Wire.h>
#include <BH1750.h>

const char* ssid = "Network Name";    // Replace with the Network Name
const char* password = "Password"; // Replace with the Network Password

const char* serverIP = "192.168.50.167";   // Replace with the IP address of the central server ESP32

BH1750 lightMeter;

void setup() {
  Serial.begin(115200);

  Wire.begin();
  lightMeter.begin();

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  float lux = lightMeter.readLightLevel();

  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    String serverPath = "http://" + String(serverIP) + "/updateLight?value=" + String(lux);
    http.begin(serverPath);
    int httpCode = http.GET();
    if (httpCode > 0) {
      String payload = http.getString();
      Serial.println(payload);
    }
    http.end();
  }
  delay(10000);  // Adjust the delay as needed
}
```

- **Arduino IDE Serial Monitor Output:**

```
Output    Serial Monitor ×
Message (Enter to send message to 'ESP3

Light: 65.00 lx
Light: 63.33 lx
Light: 67.50 lx
Light: 71.67 lx
Light: 71.67 lx
Light: 114.17 lx
Light: 118.33 lx
Light: 102.50 lx
Light: 104.17 lx
Light: 105.00 lx
Light: 106.67 lx
Light: 106.67 lx
Light: 106.67 lx
Light: 106.67 lx
Light: 105.00 lx
Light: 99.17 lx
Light: 100.83 lx
Light: 101.67 lx
Light: 101.67 lx
Light: 101.67 lx
Light: 101.67 lx
```

8.3.2 TCS3200 COLOR SENSOR:

- **Description:**

TCS3200 [50] is basically a Programable Color Light to Frequency converter, that detects the amount of R-G-B content in any light source and represents it in terms of frequency. For further details we can refer to Chapter 3.4 as discussed above.

- **Specifications:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
TCS3200	Length (mm)	32	1
	Width (mm)	25	
	Height (mm)	11	
	Weight (g)	4	
	Operating voltage (VCC)	3 V to 5V	
	Operating current	120 µA to 190 µA (max)	
	Operating Temperature	-40°C ~ 85°C	
	Detection Distance	10 mm	

➢ Component Diagram:

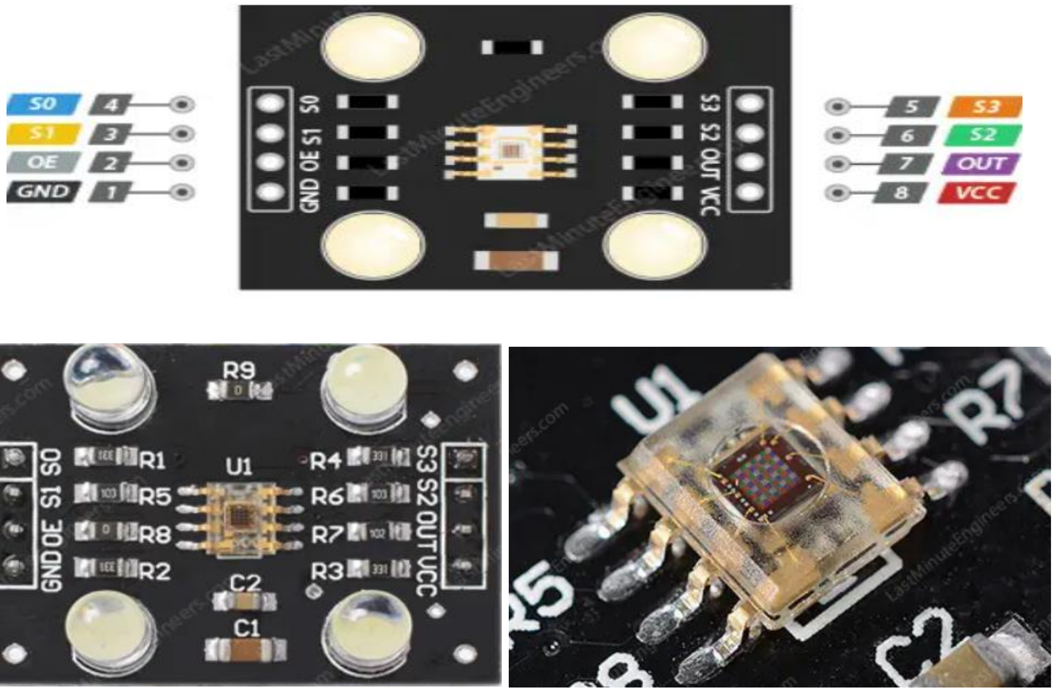


Figure 17: TCS3200 Pin-out & Overview [51]

➢ Features:

1. Color Sensor provides a readout of individual RGB color components as a digital frequency.
2. Onboard white LEDs can be gated on and off for ambient light subtraction.
3. The included 5.6mm lens provides a well-defined field of view.
4. The module is plug-compatible with Parallax motherboards.
5. The module can run from 3.3V or 5V logic supplies, plus 5V for the LED.

➢ Practical Applications:

1. TCS3200 is used to detect color of any surface.
2. It is applied in medical diagnosis. Used in industrial process control.
3. In laser edge banding machines to detect color TCS3200 is used.
4. To detect chronic kidney diseases TCS3200 is used for urine analysis.
5. TCS3200 is used in fruit sorting systems. The intensity of blue, red, green radiations can be measured using this sensor module.
6. For designing of Light-to-Frequency receiver in Multi-color visible light communication, TCS3200 is used.

➢ Interfacing TCS3200 With ESP32:

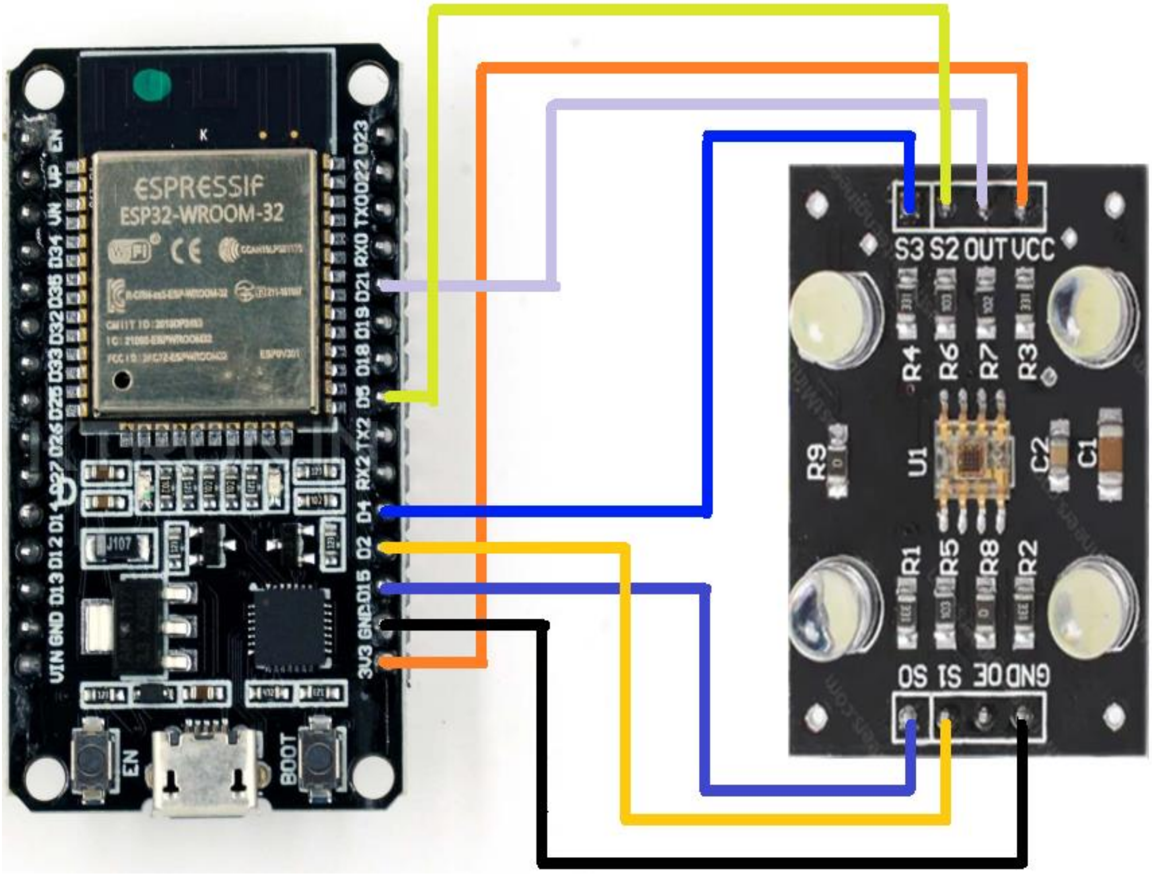


Figure 18: Connection of TCS3200 with ESP32

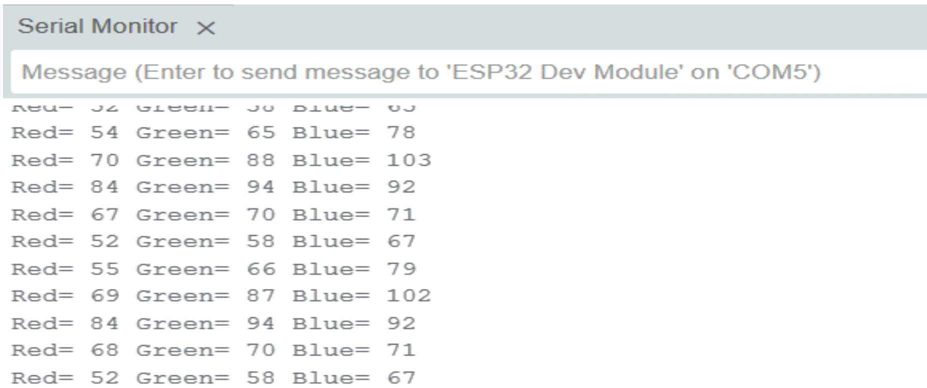
➢ Uploading Code (Arduino IDE):

```
#include <WiFi.h>
#include <WebServer.h>

WebServer server(80);

const char* ssid = "Network Name";    /*
Enter Your SSID */
const char* password = "Password"; /* Enter
Your Password */

#define S0 15
#define S1 2
#define S2 5
#define S3 4
#define sensorOut 21

int redValue;
int greenValue;
int blueValue;
float luxValue = 0;
int frequency;

void MainPage() {
  String _html_page =
"<html><body><h1>Sensor
Values</h1><p>Light: <span
id='lux'></span> lx</p><p>Red: <span
id='red'></span></p><p>Green: <span
id='green'></span></p><p>Blue: <span
id='blue'></span></p><script>setInterval(fu
nction(){fetch('/readColors').then(response
=> response.json()).then(data =>
{document.getElementById('lux').innerText
=
data[0];document.getElementById('red').inn
erText =
```

```
data[1];document.getElementById('green').i
nnerText =
data[2];document.getElementById('blue').in
nerText = data[3];});},
1000);</script></body></html>";
  server.send(200, "text/html", _html_page);
}

void Colors() {
  String data = "[\"" + String(luxValue) +
"\",\"" + String(redValue) + "\",\"" +
String(greenValue) + "\",\"" +
String(blueValue) + "\"]";
  server.send(200, "application/json", data);
}

void handleLight() {
  if (server.hasArg("value")) {
    luxValue = server.arg("value").toFloat();
  }
  server.send(200, "text/plain", "OK");
}

void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);

  // Setting frequency-scaling to 20%
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);

  Serial.begin(115200);

  WiFi.mode(WIFI_STA);        /* Set the
WiFi in STA Mode */
  WiFi.begin(ssid, password);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  delay(1000);              /* Wait for 1000ms
*/
  while (WiFi.status() !=
WL_CONNECTED) {
    Serial.print(".");
    delay(200);
  } /* Wait while connecting to WiFi */
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("Your Local IP address is: ");
  Serial.println(WiFi.localIP());       /* Print
the Local IP */

  server.on("/", MainPage);             /*
Display the Web/HTML Page */
  server.on("/readColors", Colors);     /*
Display the updated RGB values */
  server.on("/updateLight", handleLight); /*
Handle light sensor updates */
  server.begin();                       /* Start Server
*/
  delay(1000);                          /* Wait for
1000ms */
}

void loop() {
  redValue = getRed();   /* Get the Red
Color Value */
  delay(100);
  greenValue = getGreen(); /* Get the Green
Color Value */
  delay(100);
  blueValue = getBlue();   /* Get the Blue
Color Value */
  delay(100);

  Serial.print("Red = ");
  Serial.print(redValue);   /* Print Red Color
Value on Serial Monitor */
  Serial.print("    ");
  Serial.print("Green = ");
  Serial.print(greenValue); /* Print Green
Color Value on Serial Monitor */
  Serial.print("    ");
  Serial.print("Blue = ");
  Serial.println(blueValue);/* Print Blue
Color Value on Serial Monitor */

  server.handleClient();
  delay(1000);             /* wait a second */
}

int getRed() {
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
```

```
  frequency = pulseIn(sensorOut, LOW); /* Get the Red Color Frequency */
  return frequency;
}

int getGreen() {
  digitalWrite(S2, HIGH);
  digitalWrite(S3, HIGH);
  frequency = pulseIn(sensorOut, LOW); /* Get the Green Color Frequency */
  return frequency;
}

int getBlue() {
  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  frequency = pulseIn(sensorOut, LOW); /* Get the Blue Color Frequency */
  return frequency;
}
```

- **Arduino IDE Serial Monitor Output:**



8.3.3 ESP32 MODULE:

- **Description:**

ESP32 [52] is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC low-power 40 nm technology.

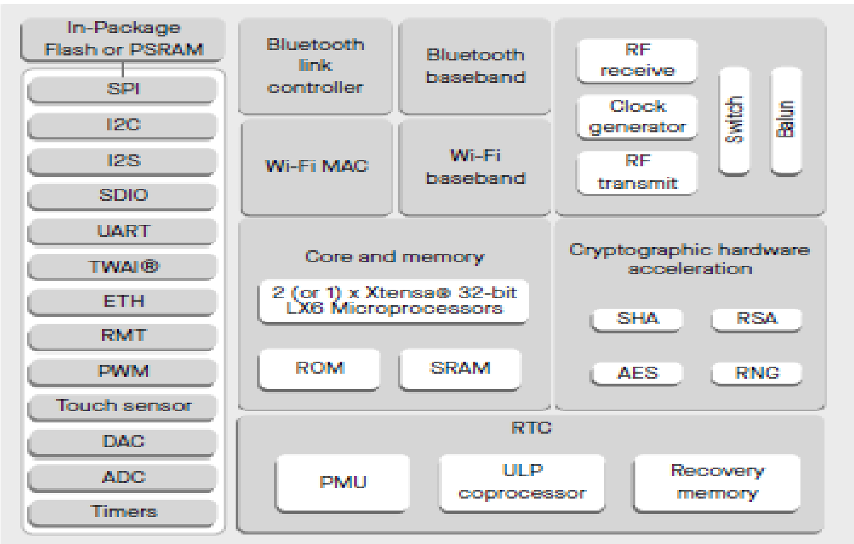


Figure 19: ESP32 Functional Block Diagram [52]

It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios. The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.

- **Specifications:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
ESP32 Development Module	Microprocessor	Dual-Core 32-bit	2
	Clock Frequency	240 MHz	
	Bandwidth	72 MHz	
	SRAM	520 KB	
	ROM	448 KB	
	RTC SRAM	16 KB	
	Interface	Ethernet, I2C, I2S, SPI, UART	
	Operating Temperature	-40 to 85 ºC	
	Supply Voltage	3 V to 3.6 V	
	Supply Current	500 mA	
	Wi-Fi Speed	150 Mbps	
	Bluetooth	v4.2	
	Programmable GPIOs	34	
	12-bit SAR ADC	18 Channels	
	8-bit DAC	2 Channels	

- **Component Diagram:**

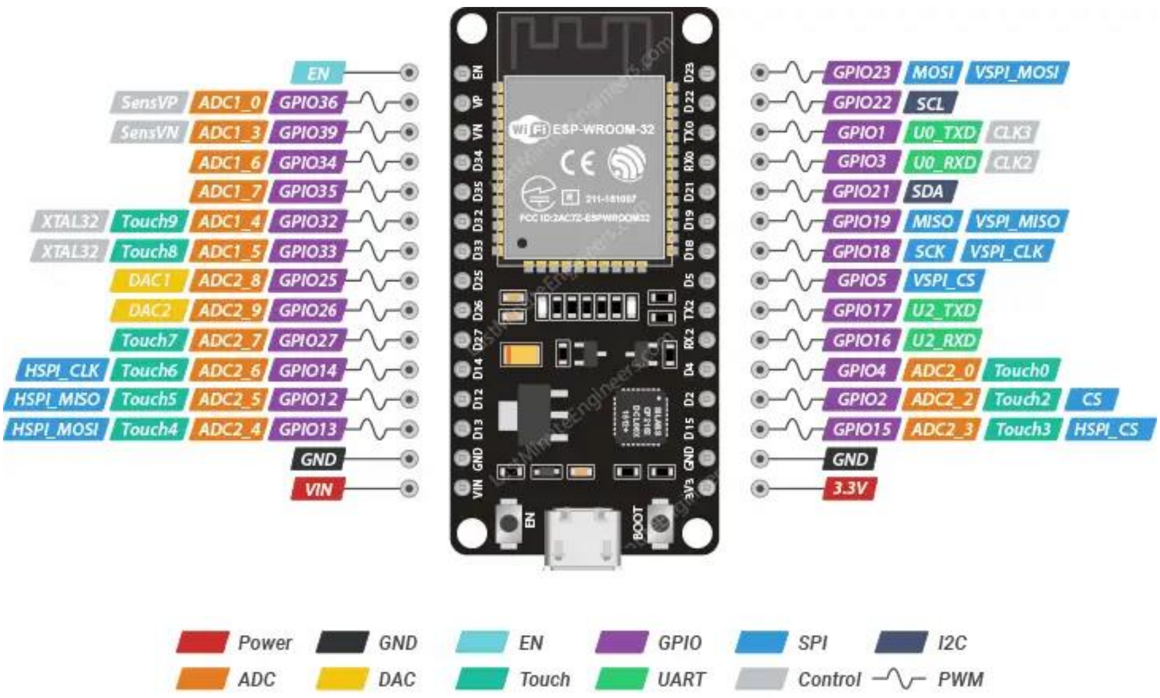


Figure 20: ESP32 Pin-Out Diagram [53]

- **Features:**

1. Compliant with Bluetooth v4.2 BR/EDR and Bluetooth LE specifications.
2. Enhanced Power Control, +9 dBm transmitting power, high-speed UART HCI, up to 4 Mbps.
3. Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes and antenna diversity.
4. Fine-resolution power control through a selection of clock frequency, duty cycle, Wi-Fi operating modes, and individual power control of internal components
5. Five power modes designed for typical scenarios: Active, Modem-sleep, Light-sleep, Deep-sleep, Hibernation

- **Practical Applications:**

1. Smart Home, Industrial Automation, IoT, Health care.
2. Smart Agriculture, Speech Recognition, Image Processing.
3. Service Robot, Audio Devices etc.

- **ESP32 Board Construction:**

Referring to the Figure 22 we can name the different parts of ESP32 Board as given below:

1. ESP-WROOM-32 Module
2. Two rows of IO Pins (with 15 pins on each side)

3. CP2012 USB – UART Bridge IC
4. micro–USB Connector (for power and programming)
5. AMS1117 3.3V Regulator IC
6. Enable Button (for Reset)
7. Boot Button (for flashing)
8. Power LED (Red)
9. User LED (Blue – connected to GPIO2)

➢ **Programming Environment:**

Some of the common platform to program ESP32 Module for different operations are:

1. Arduino IDE
2. PlatformIO IDE (VS Code)
3. LUA
4. MicroPython
5. Espressif IDF (IoT Development Framework)
6. JavaScript

8.3.4 VERO BOARD:

➢ **Description:**

In this project to simulate the PCB circuit we need Vero Board. It is the simplest way to design circuit using soldering. We can refer to the "model overview" for our better understanding.

➢ **Specifications:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
Vero Board	Length (mm)	150	1
	Width (mm)	90	
	Height (mm)	2	
	Weight (g)	75	
	Hole Pitch (mm)	2.54	
	Built Material	Fiber Glass	

- **Component Diagram:**

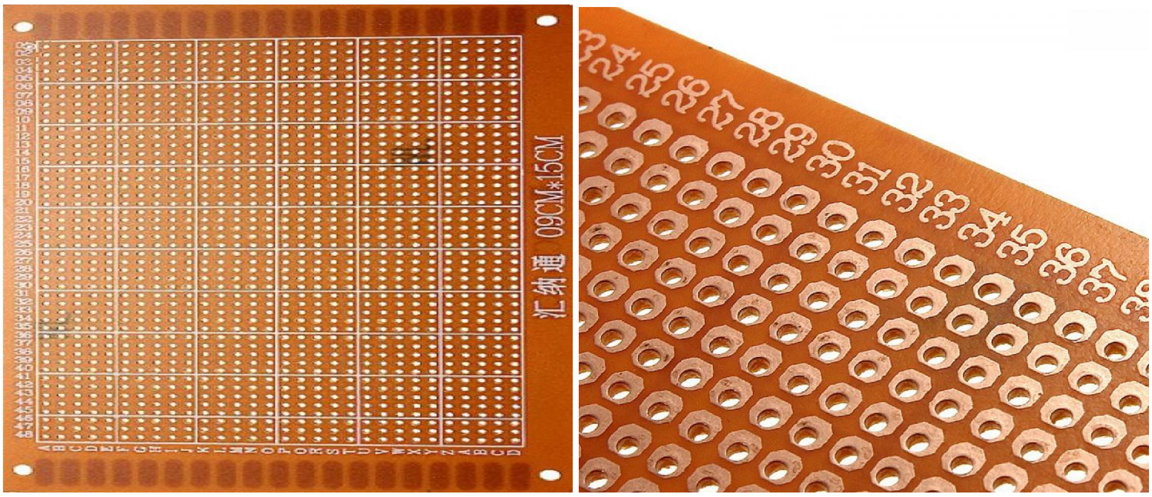


Figure 21: Overview of Vero Board

8.3.5 SINGLE STRANDED WIRES:

- **Description:** Single-stranded wire is a type of electrical wire that consists of a single conducting wire core, typically made of copper or aluminium. It is used in various electrical and electronic applications where flexibility and ease of installation are important. Single-stranded wire is known for its simplicity and cost-effectiveness.
- **Specifications:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
Single Stranded Wire	Length (mm)	100	1
	Cable Gauge	23 AWG	
	Voltage	150 V	
	Operating Temperature	-10 to 80 ºC	
	Max. DC resistance	63.5Ω/km at 20°C	

- **Component Diagram:**

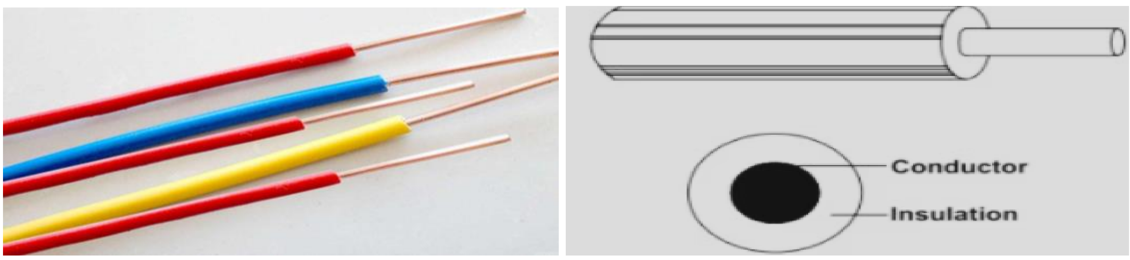


Figure 22: Single Stranded Wire Overview

8.4 METHODOLOGY

The Integrated Sensor set-up can measure Illuminance level and R-G-B values, but on the basis of some particular methodologies. This section outlines the methodology used to integrate sensor data from two ESP32 microcontroller units, each connected to different sensors, and aggregate the data on a single webpage. The project involved setting up one ESP32 as a central server hosting the webpage and another ESP32 as a client that sends its sensor data to the server. For this purpose, we need a stable Wi-Fi Connection [61] such that we could connect two ESP-32 with a particular Wi-Fi Network.

System Design: The system was designed such that;

Central Server ESP32: Connected to the TCS3200 color sensor, it hosts a webpage displaying the sensor values.

Client ESP32: Connected to the BH1750 light sensor, it reads the light intensity and sends this data to the central server via HTTP requests.

- **Methodological Steps:**

1. **Configuring the Central Server ESP32 (TCS3200):** Firstly, we can refer to the coding of Chapter 8.3.2 and we have to upload the code using Arduino IDE. After that, we have to open the Serial Monitor output to note the IP address assigned to the ESP32 by the WiFi network. In the Serial Monitor, we will see messages indicating the ESP32 is attempting to connect to the specified WiFi network. Once connected, the Serial Monitor will display **Connected to <Your SSID>** followed by **Your Local IP address is: <IP Address>**. Now we have to note down the local IP address assigned to our ESP32 (e.g., 192.168.1.10).
2. **Configuring the Client ESP32 (BH1750):** Firstly, we can refer to the coding of Chapter 8.3.1 and upload the code using Arduino IDE. Then we have to monitor the data displayed in the webpage screen.
3. **Access the Web Page:** Then, we will open a web browser on a device connected to the same WiFi network as our ESP32. After that, we have to enter the local IP address noted from the Serial Monitor into the address bar (e.g., http://192.168.1.10). Then, the browser should display a web page with headings for **Light** and **Red**, **Green**, and **Blue** color values.
4. **Webpage Interaction:** The webpage [60] will display real-time values from both the TCS3200 and BH1750 sensors. The page automatically updates every second to reflect

the latest sensor readings. This is handled by a JavaScript script embedded in the HTML, which fetches updated sensor data from the server.

5. **Testing and Validation:** Now, we have to place the TCS3200 sensor under various coloured lights and the BH1750 sensor in different lighting conditions. And also, we have to observe the values from webpage. If accurate measurements are seemed to be coming then, the system can be deployed for continuous monitoring of color and light intensity.

we can refer to the below picture, where we can see the reading at a particular IP Address of the Webpage. This webpage continuously refreshes itself, and provides a real-time analysis of the R-G-B and light intensity of the surrounding lighting environment.

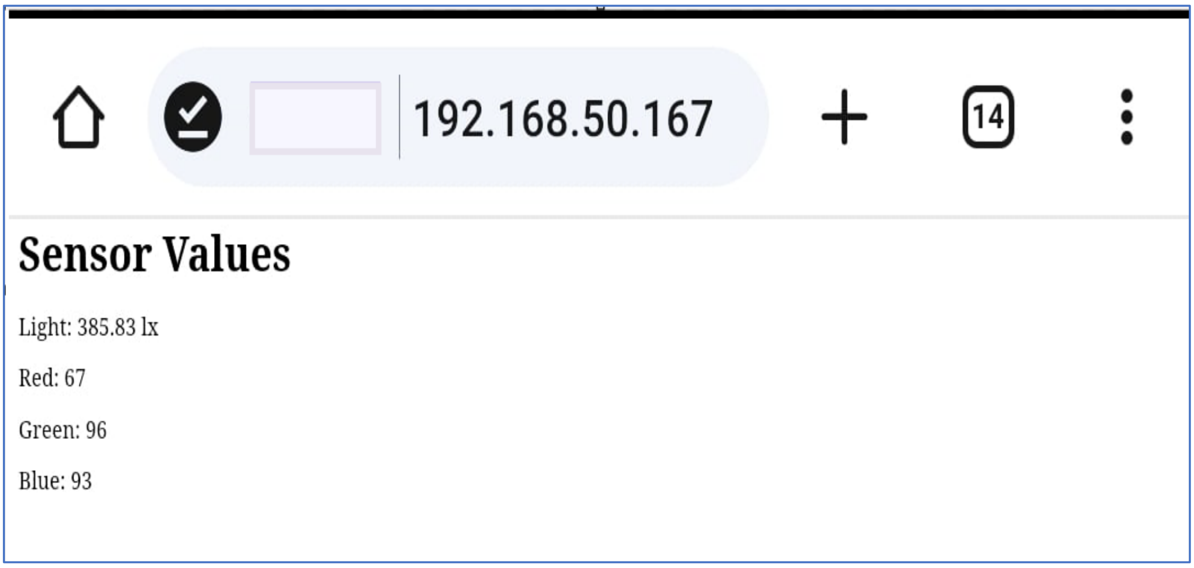


Figure 23: Webpage Output Screen of The Sensors

8.5 PREPARATION OF LOOK-UP TABLE

Now after calibrating the RGB Sensors, we used the Color Tunable LED Smart Blub to prepare the Look-Up table. This table is necessary to compute the CCT or to estimate the CCT value of test light source based on its red, green and blue responses. This table will be tabulated in an excel sheet and using python coding we will call that excel sheet as a matrix and we will compute the CCT.

➢ **Methodology:**

1. Firstly, we will connect the Philips Smart Wiz LED bulb to android devices.
2. Then we will vary the Correlated Color Temperature within the range of **2400 K** to **5500 K**
3. By placing the RGB Sensor in the nadir point of the black box and by varying the CCT of the light source to different values, we have to record the Red (R), Green (G), Blue (B) values.
4. Thus, for a particular CCT value we will record R, G, B values and thus table has been prepared.
5. Now, whenever we will measure the R, G and B values of any test light source from sensors the computer programming will estimate the CCT values based on the Table that we have prepared.

➢ **Look-Up Table:**

CCT (Kelvin)	Red (R)	Green (G)	Blue (B)
2400	35	91	108
2700	44	57	67
3000	46	57	67
3200	43	54	65
3400	41	52	60
3600	41	48	60
3900	40	48	59
4100	43	53	63
4200	42	51	60
4300	39	46	57
4500	39	47	56
4800	40	47	57
5100	37	44	46
5300	36	43	44

CHAPTER 9: SYSTEM DEVELOPMENT PART II : CCT COMPUTATION

From the conceptual analysis of the theoretical computation of CCT we have implemented a software coding (Python). This chapter describes the algorithm to compute CCT and it also describes the features and challenges of that particular code. So from the previous section after interfacing the sensors we can start the measurement. Now through Arduino IDE we will get the required R, G and B responses. Now to compute the CCT value from that frequency responses of RGB values, we will follow the below described algorithm.

9.1 ALGORITHM FOR CCT COMPUTATION

Serial Monitor Data Monitoring

Read Excel Data

Form Matrix

Define Function to Find Nearest Values

Calculate Nearest Values

Calculate Coefficients

Define Function to Find Nearest CCT Value

Calculate CCT Values

Print CCT Values

Final Calculation for CCT

9.2 SERIAL MONITOR RGB DATA MONITORING

In this section we will see the required process that need to be followed to monitor the RGB responses data through the serial monitor. Now for that, there are certain steps to be followed; which are as discussed as below.

- **Steps:**

1. First, we have to run the Arduino IDE code connecting the sensors.
2. Then we have to verify in the serial monitor of the Arduino IDE, whether output value is showing or not.
3. Then we have to run the below code through any Python execution software.
4. Finally, we will see that the serial monitor output is connected and data is printed in the output of Visual Studio Code (Python Software).

- **Input Code:**

```
import pandas as pd

# Read the Excel file into a Data Frame
file_path = 'D:\SOHAM SAIN\RGB Functions.xlsx'
df = pd.read_excel(file_path)

# Form a matrix from specific columns
matrix = df[['CCT', 'Red', 'Green', 'Blue']].values

# Display the matrix
print(matrix)

import serial

# Define the COM port and baud rate
ser = serial.Serial('COM5', 115200)  # Replace 'COM5' with the correct COM
port

try:
    while True:
        # Read data from the serial port
        data = ser.readline().decode('utf-8')
        if data:
            print(data, end='')  # Print the data from the Arduino
except KeyboardInterrupt:
    ser.close()  # Close the serial connection on Ctrl+C
```

9.3 PYTHON CODE FOR CCT COMPUTATION

Now after interfacing the serial monitor with the python software, we can finally calculate the CCT values by following the below steps and through execution of the code.

- **Steps:**

1. First, we have to analyze the values of RGB responses and then have to set target values of R, G and B for which CCT needs to be calculated.
2. Now we have to write the below code in any python language software.
3. Then we have to copy the directory path of the excel file where the look-up table have been created.
4. Then we have to set the target values of R, G and B in the required area.
5. Finally, by running the code we will get the required CCT values.

- **Input Code:**

```
import pandas as pd
# Read the Excel file into a
DataFrame
file_path = 'D:\SOHAM SAIN\RGB
Functions.xlsx'
df = pd.read_excel(file_path)

# Form a matrix from specific
columns
matrix = df[['CCT', 'Red', 'Green',
'Blue']].values

# Display the matrix
print(matrix)
import serial
import re

def find_nearest_values(matrix,
columns, targets):
nearest_values = {}

for column_name in columns:
nearest_values[column_name] = []

for column_name, target_value in
zip(columns, targets):
column = matrix[column_name]

nearest_value1 = None
nearest_value2 = None
nearest_difference1 = float('inf')
nearest_difference2 = float('inf')

for i in range(1, len(column)):
current_value = column[i]
previous_value = column[i - 1]

# Calculate the absolute differences
diff_current = abs(current_value -
target_value)
diff_previous = abs(previous_value -
target_value)

if diff_current <
nearest_difference1:
nearest_difference2 =
nearest_difference1
nearest_difference1 = diff_current
nearest_value2 = nearest_value1
nearest_value1 = current_value
elif diff_current <
nearest_difference2:
nearest_difference2 = diff_current
nearest_value2 = current_value

if diff_previous <
nearest_difference1:
nearest_difference2 =
nearest_difference1
nearest_difference1 = diff_previous
nearest_value2 = nearest_value1
nearest_value1 = previous_value
elif diff_previous <
nearest_difference2:
```

```
nearest_difference2 = diff_previous
nearest_value2 = previous_value

nearest_values[column_name] =
(nearest_value1 - 2, nearest_value2
+ 1)  # Append as a tuple, not a
list

return nearest_values

matrix = pd.DataFrame({'CCT': [2400,
2700, 3000, 3200, 3400, 3600, 3900,
4100, 4200, 4300, 4500, 4800, 5100,
5300],
'Red': [35, 44, 46, 43, 41, 41, 40,
43, 42, 39, 39, 40, 37, 36],
'Green': [97, 57, 57, 54, 52, 48,
48, 53, 51, 46, 47, 47, 44, 43],
'Blue': [108, 67, 67, 65, 60, 60,
59, 63, 60, 57, 56, 57, 46, 44]})

# Specify target values for each
column
target_values = [44, 53, 64]
column_names = ['Red', 'Green',
'Blue']
nearest_values =
find_nearest_values(matrix,
column_names, target_values)
print("Nearest values to targets
are:")
for column_name in nearest_values:
print(f"For {column_name}:
{nearest_values[column_name]}")
k11, k12, k13, R2, R3= 0, 0, 0, 0, 0
# For RED Value
R = target_values[0]  # Access the
first value (44) from the
target_values list
R1 = R
red_nearest_values =
nearest_values['Red']
r2, r3 = red_nearest_values
R2, R3= r2+4, r3-2
k11 = ((R - R2) * (R - R3)) / ((R1 -
R2) * (R1 - R3))
k12 = ((R - R1) * (R - R3)) / ((R2 -
R1) * (R2 - R3))
k13 = ((R - R1) * (R - R2)) / ((R3 -
R1) * (R3 - R2))
print(R1)
print("Value of R2:", R2)
print("Value of R3:", R3)
print("Value of K11, K12, K13:",
k11, k12, k13)
k21, k22, k23, G2, G3 = 0, 0, 0, 0,
0
# For Green Value
G = target_values[1]  #Access the
second value (53) from the
target_values list
G1 = G + 4
green_nearest_values =
nearest_values['Green']
g2, g3 = green_nearest_values
G2, G3 = g2 + 5, g3
k21 = ((G - G2) * (G - G3)) / ((G1 -
G2) * (G1 - G3))
k22 = ((G - G1) * (G - G3)) / ((G2 -
G1) * (G2 - G3))
k23 = ((G - G1) * (G - G2)) / ((G3 -
G1) * (G3 - G2))
print(G1)
print("Value of G2:", G2)
print("Value of G3:", G3)
print("Value of K21, K22, K23:",
k21, k22, k23)

k31, k32, k33, B2, B3 = 0, 0, 0, 0,
0
# For Blue Value
B = target_values[2]  # Access the
third value (64) from the
target_values list
B1 = B + 3
blue_nearest_values =
nearest_values['Blue']
b2, b3 = blue_nearest_values
B2, B3 = b2 + 3, b3 - 1
k31 = ((B - B2) * (B - B3)) / ((B1 -
B2) * (B1 - B3))
k32 = ((B - B1) * (B - B3)) / ((B2 -
B1) * (B2 - B3))
k33 = ((B - B1) * (B - B2)) / ((B3 -
B1) * (B3 - B2))
print(B1)
print("Value of B2:", B2)
print("Value of B3:", B3)
print("Value of K31, K32, K33:",
k31, k32, k33)
```

```
def find_nearest_cct_value(matrix,
red, green, blue):
min_distance = float('inf')
nearest_cct = None
for _, row in matrix.iterrows():
cct = row['CCT']
r = row['Red']
g = row['Green']
b = row['Blue']
# Calculate the Euclidean distance
distance = ((r - red) ** 2 + (g -
green) ** 2 + (b - blue) ** 2) **
0.5
if distance < min_distance:
min_distance = distance
nearest_cct = cct
return nearest_cct

# Get the nearest CCT values for the
obtained Red, Green, and Blue values
cct_red =
find_nearest_cct_value(matrix, R1,
G1, B1)
cct_green =
find_nearest_cct_value(matrix, R2,
G2, B1)
cct_blue =
find_nearest_cct_value(matrix, R3,
G3, B1)
print(f"CCT value for Red (R1, G1,
B1): {cct_red}")
print(f"CCT value for Green (R2, G2,
B1): {cct_green}")
print(f"CCT value for Blue (R3, G3,
B1): {cct_blue}")

#Final Calculation For CCT
C = k21*cct_red + k22*cct_green +
k23*cct_blue
print("Correlated Colour Temperature
=",C,"Kelvin")
```

- **Output:**

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

[[2400   35   91  108]
 [2700   44   57   67]
 [3000   46   57   67]
 [3200   43   54   65]
 [3400   41   52   60]
 [3600   41   48   60]
 [3900   40   48   59]
 [4100   43   53   63]
 [4200   42   51   60]
 [4300   39   46   57]
 [4500   39   47   56]
 [4800   40   47   57]
 [5100   37   44   46]
 [5300   36   43   44]]
Nearest values to targets are:
For Red: (42, 45)
For Green: (51, 54)
For Blue: (63, 66)
44
Value of R2: 46
Value of R3: 43
Value of K11, K12, K13: 1.0 0.0 0.0
57
Value of G2: 56
Value of G3: 54
Value of K21, K22, K23: 1.0 -2.0 2.0
67
Value of B2: 66
Value of B3: 65
Value of K31, K32, K33: 1.0 -3.0 3.0
CCT value for Red (R1, G1, B1): 2700
CCT value for Green (R2, G2, B1): 3000
CCT value for Blue (R3, G3, B1): 3200
Correlated Colour Temperature = 3100.0 Kelvin
PS C:\Users\soham\OneDrive\Desktop\Python Code>
```

9.4 ANALYSIS OF THE CODING

The above coding has been analyzed in detail through different sections of the coding. It's as given below:

9.4.1 READING EXCEL DATA & MATRIX FORMATION:

```
import pandas as pd

# Read the Excel file into a DataFrame
file_path = 'D:\SOHAM SAIN\RGB Functions.xlsx'
df = pd.read_excel(file_path)
# Form a matrix from specific columns
matrix = df[['CCT', 'Red', 'Green', 'Blue']].values
# Display the matrix
print(matrix)
```

➢ **Working:**

1. This section reads an Excel file containing data related to CCT (Correlated Color Temperature), Red, Green, and Blue values.
2. It uses Pandas to read the Excel file into a Data Frame.
3. Then, it selects specific columns ('CCT', 'Red', 'Green', 'Blue') and converts them into a NumPy array called matrix.
4. Finally, it prints the matrix.

9.4.2 DEFINING FUNCTION TO FIND NEAREST VALUES:

```
def find_nearest_values(matrix, columns, targets):
nearest_values = {}

for column_name in columns:
nearest_values[column_name] = []

for column_name, target_value in zip(columns, targets):
column = matrix[column_name]

nearest_value1 = None
nearest_value2 = None
nearest_difference1 = float('inf')
nearest_difference2 = float('inf')

for i in range(1, len(column)):
current_value = column[i]
previous_value = column[i - 1]

# Calculate the absolute differences
diff_current = abs(current_value - target_value)
```

```
diff_previous = abs(previous_value - target_value)

if diff_current < nearest_difference1:
nearest_difference2 = nearest_difference1
nearest_difference1 = diff_current
nearest_value2 = nearest_value1
nearest_value1 = current_value
elif diff_current < nearest_difference2:
nearest_difference2 = diff_current
nearest_value2 = current_value
if diff_previous < nearest_difference1:
nearest_difference2 = nearest_difference1
nearest_difference1 = diff_previous
nearest_value2 = nearest_value1
nearest_value1 = previous_value
elif diff_previous < nearest_difference2:
nearest_difference2 = diff_previous
nearest_value2 = previous_value

nearest_values[column_name] = (nearest_value1 - 2, nearest_value2 + 1)  # Append as a tuple, not a list

return nearest_values
```

➢ **Working:**

1. The function **find_nearest_values** takes three arguments: matrix, columns, and targets.
2. It calculates the nearest values to the target values for each column in the matrix.
3. For each column, it iterates through the values and finds the two nearest values to the target value.
4. It stores these two nearest values in a dictionary **nearest_values**.
5. Finally, it returns the **nearest_values** dictionary.

9.4.3 CALCULATING NEAREST VALUES & COEFFICIENTS:

```
matrix = pd.DataFrame({'CCT': [2400, 2700, 3000, 3200, 3400, 3600, 3900, 4100, 4200, 4300, 4500, 4800, 5100, 5300],
'Red': [35, 44, 46, 43, 41, 41, 40, 43, 42, 39, 39, 40, 37, 36],
'Green': [97, 57, 57, 54, 52, 48, 48, 53, 51, 46, 47, 47, 44, 43],
'Blue': [108, 67, 67, 65, 60, 60, 59, 63, 60, 57, 56, 57, 46, 44]})

# Specify target values for each column
target_values = [44, 53, 64]
column_names = ['Red', 'Green', 'Blue']
nearest_values = find_nearest_values(matrix, column_names, target_values)

print("Nearest values to targets are:")
for column_name in nearest_values:
```

```
print(f"For {column_name}: {nearest_values[column_name]}")

k11, k12, k13, R2, R3= 0, 0, 0, 0, 0
# For RED Value
R = target_values[0]  # Access the first value (46) from the target_values
list
R1 = R
red_nearest_values = nearest_values['Red']
r2, r3 = red_nearest_values
R2, R3= r2+4, r3-2
k11 = ((R - R2) * (R - R3)) / ((R1 - R2) * (R1 - R3))
k12 = ((R - R1) * (R - R3)) / ((R2 - R1) * (R2 - R3))
k13 = ((R - R1) * (R - R2)) / ((R3 - R1) * (R3 - R2))
print(R1)
print("Value of R2:", R2)
print("Value of R3:", R3)

print("Value of K11, K12, K13:", k11, k12, k13)

#Similar Calculation for Green and Blue
```

- **Working:**

1. This section first creates a sample matrix for testing purposes.
2. Then, it specifies target values for each column ('Red', 'Green', 'Blue').
3. It calls the previously defined function find_nearest_values to find the nearest values for each target.
4. After that, it calculates coefficients (k11, k12, k13, k21, k22, k23, k31, k32, k33) for each color component (Red, Green, Blue) based on the nearest values.

9.4.4 DEFINING FUNCTIONS TO FIND NEAREST CCT VALUE:

```
def find_nearest_cct_value(matrix, red, green, blue):
min_distance = float('inf')
nearest_cct = None

for _, row in matrix.iterrows():
cct = row['CCT']
r = row['Red']
g = row['Green']
b = row['Blue']
# Calculate the Euclidean distance
distance = ((r - red) ** 2 + (g - green) ** 2 + (b - blue) ** 2) ** 0.5

if distance < min_distance:
min_distance = distance
nearest_cct = cct
return nearest_cct
```

- **Working:**

1. This function **find_nearest_cct_value** takes four arguments: matrix, red, green, and blue.
2. It iterates over each row of the matrix.
3. For each row, it calculates the Euclidean distance between the RGB values of the row and the given RGB values.
4. It updates **min_distance** and **nearest_cct** if a closer CCT value is found.
5. Finally, it returns the nearest CCT value.

9.4.5 CALCULATING FINAL CCT VALUE:

```
# Get the nearest CCT values for the obtained Red, Green, and Blue values
cct_red = find_nearest_cct_value(matrix, R1, G1, B1)
cct_green = find_nearest_cct_value(matrix, R2, G2, B1)
cct_blue = find_nearest_cct_value(matrix, R3, G3, B1)

print(f"CCT value for Red (R1, G1, B1): {cct_red}")
print(f"CCT value for Green (R2, G2, B1): {cct_green}")
print(f"CCT value for Blue (R3, G3, B1): {cct_blue}")

#Final Calculation For CCT
C = k21*cct_red + k22*cct_green + k23*cct_blue
print("Correlated Colour Temperature =",C,"Kelvin")
```

- **Working:**

1. This section calculates the nearest CCT values for the obtained Red, Green, and Blue values using the function **find_nearest_cct_value**.
2. It prints the CCT values for each color component.
3. Finally, it calculates the correlated color temperature (C) using the weighted contributions of the individual color temperatures for Red, Green, and Blue components and prints the result.
4. Thus, CCT is being calculated in unit of Kelvin.

9.5 ADVANTAGES & CHALLENGES

In any scientific experimentation and product development, there are certain key features which are considered as an advantage of that model where as some of its demerits are considered as a way to further improvement. In this section we will give an idea about the key advantages and challenges of the technical coding part of this project.

- **Advantages:**

1. It reads data from an Excel file and forms a matrix.

2. It finds the nearest values and calculates coefficients for each color component.

3. It calculates the nearest CCT values for given RGB values.

4. Finally, it computes the correlated color temperature based on the weighted contributions of the individual color temperatures.

- **Challenges:**

1. The code depends on an external Excel file ('RGB Functions.xlsx'). If this file is moved or renamed, the code will fail. Ensuring robustness against such changes would require more sophisticated file handling.

2. The code lacks modularity, making it difficult to reuse or maintain. Breaking it into smaller, reusable functions with clear responsibilities would improve readability and maintainability.

3. The algorithm used to find nearest values (**find_nearest_values** function) iterates over the entire dataset multiple times, which could be inefficient for large datasets. Optimizing this algorithm for better performance, perhaps by using more efficient search algorithms, could be beneficial.

CHAPTER 10: SYSTEM DEVELOPMENT PART III : BLUETOOTH CONTROLLED CAR SET-UP

In this chapter we will discuss about the final stage of the prototype where we will implement the driving mechanism which is more likely a Bluetooth control EV set-up which will ease the movement of the prototype. The detailed overview of this part of the system development is as discussed below.

10.1 SYSTEM OVERVIEW

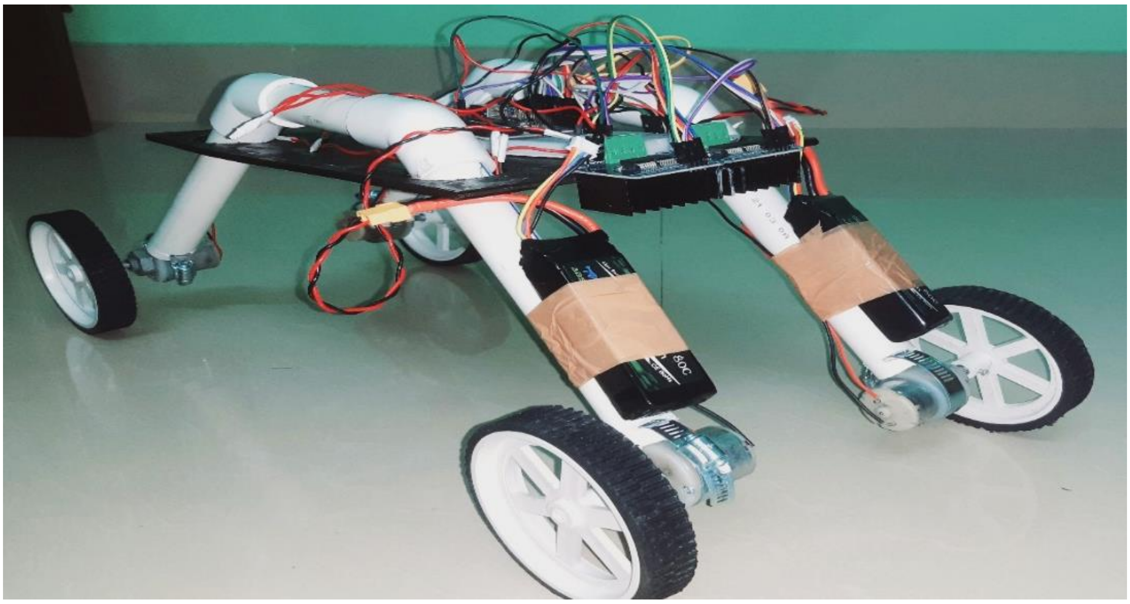


Figure 24: Front View of The Model

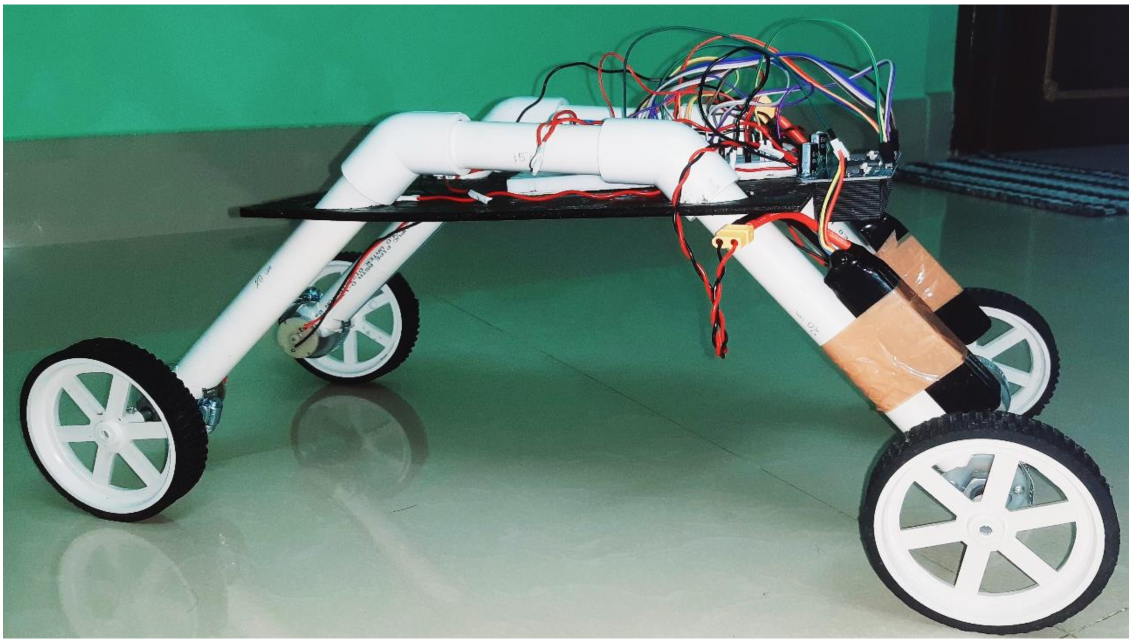
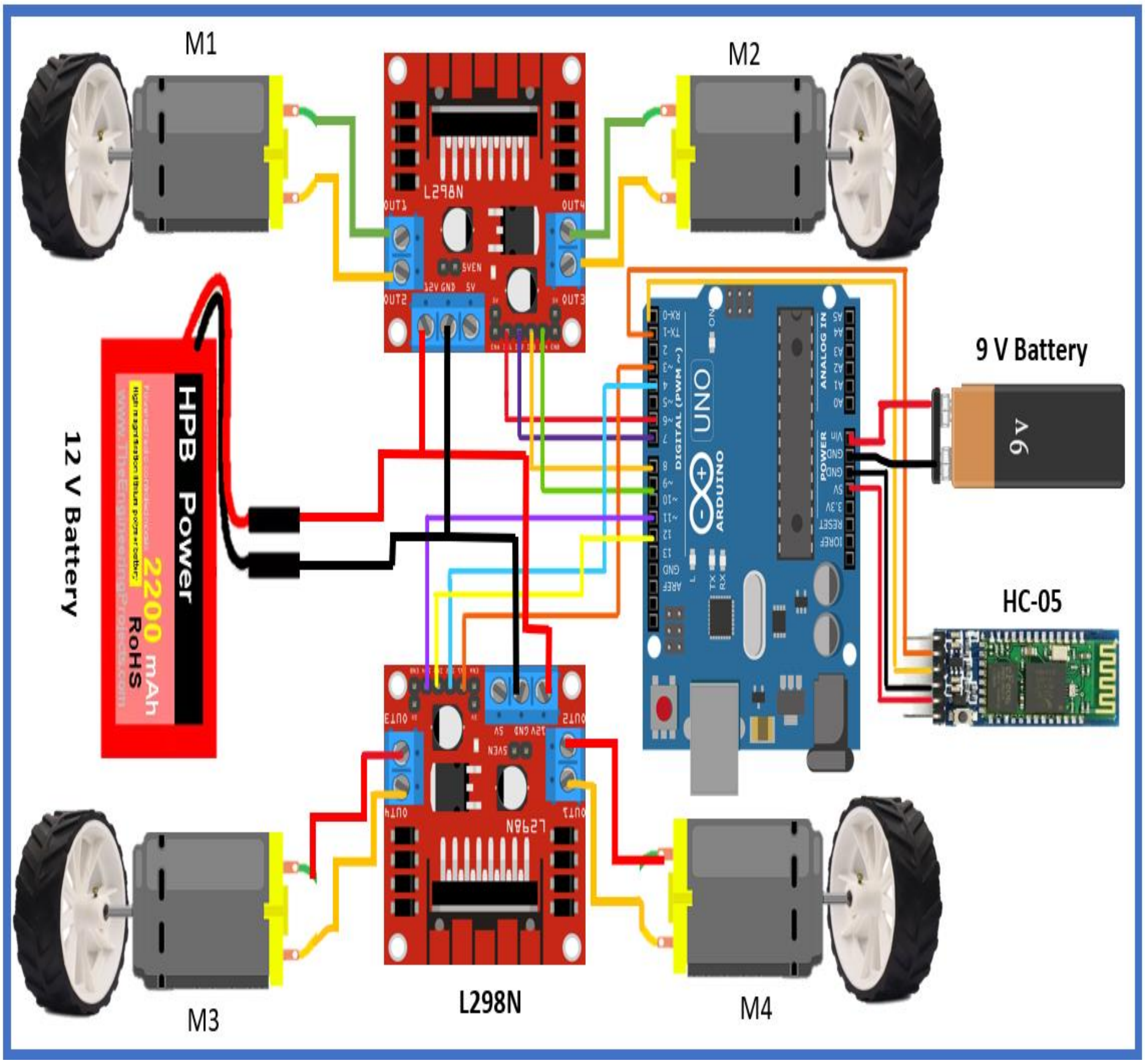
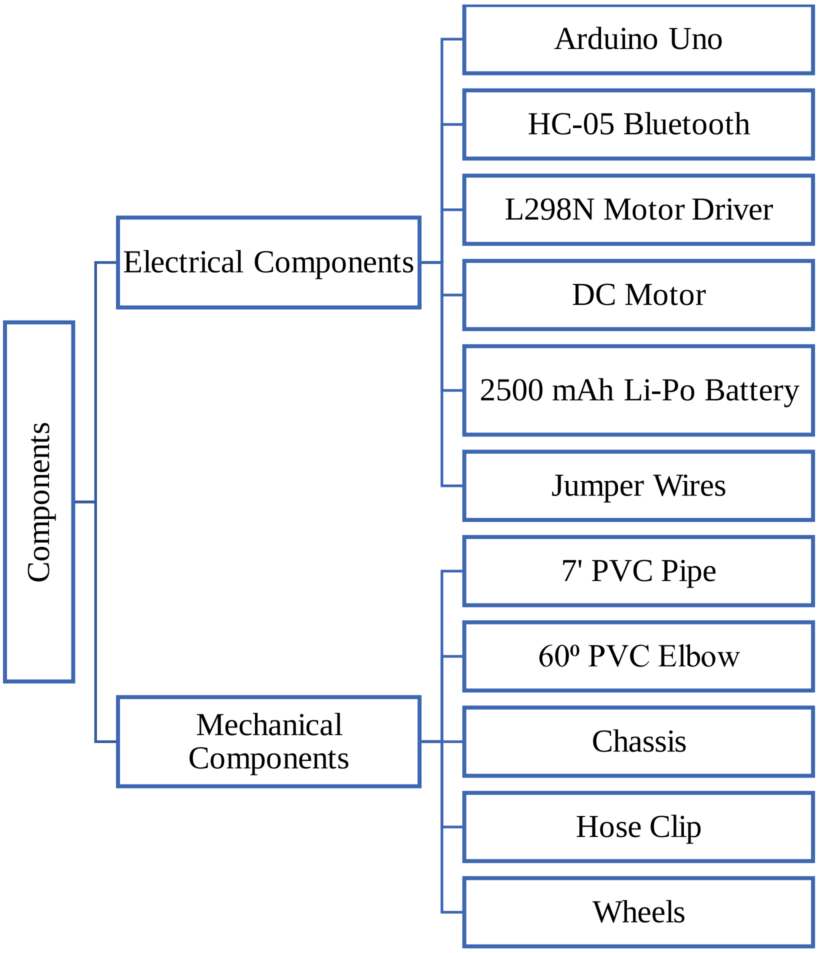


Figure 25: Side View of The Model

10.2 HARDWARE CONNECTION



10.3 COMPONENTS WITH SPECIFICATIONS



The specification of each component along with their key features have been listed below. In the above picture we can see that different components has been used in implementing the driving mechanism of the model. Each component has their own significance which has been as discussed below as well.

10.3.1 ARDUINO UNO:

➢ **Description:**

It is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output. This board contains a USB interface which is used to program the board. The user writes the Arduino code in the IDE, then uploads it to the microcontroller which executes the code, interacting with inputs and outputs such as sensors, motors, and lights.

➢ Specifications:

NAME	PARAMETER	SPECIFICATION	QUANTITY
Arduino Uno R3	Length (mm)	68.6	1
	Width (mm)	53.4	
	Height (mm)	2	
	Weight (g)	25	
	Operating voltage (V_{CC})	5V	
	Input Voltage (V_{in})	7 to 12 Volt	
	Built-in LED Pins	13	
	Digital I/O Pins	14	
	PWM Digital I/O Pins	6	
	Analog I/O Pins	6	
	DC Current per I/O Pin	20 mA	
	Clock Speed	16 MHz	
	Memory	2KB SRAM, 32KB FLASH, 1KB EEPROM	

➢ Component Diagram:

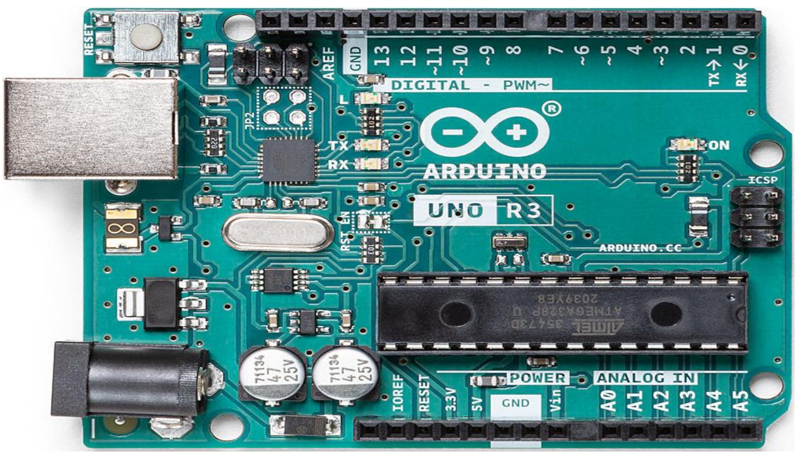


Figure 26: Simulated Diagram of Arduino Uno R3 [54]

➢ **Features:**

1. ATMega328P Processor is present in Arduino Uno R3.
2. It has 14 digital input/output pins (of which 6 can be used as PWM outputs) and 6 analog inputs. These pins allow the Uno to interface with a wide range of sensors, actuators, and other electronic components.
3. The Uno features a USB Type-B connector, which allows it to be easily connected to a computer for programming and serial communication. It can be powered via USB or an external power supply.
4. There is a built-in LED connected to digital pin 13, which can be used for simple debugging or as a visual indicator in your projects.
5. The Arduino Uno R3 is part of the Arduino ecosystem, which is open-source. This means that the hardware schematics and software libraries are freely available, allowing users to modify and extend the functionality of the board as needed.

➢ **Application:**

Some of the commonly known applications of Arduino Uno R3 are weighing machines, traffic light count down timer, parking lot counter, embedded systems, medical instrument, emergency light for railways, home & industrial automation. Moreover, many electronics projects and experiments requires use of Arduino uno.

10.3.2 HC-05 BLUETOOTH MODULE:

➢ **Description:**

The HC-05 is a Class 2 Bluetooth Module [55] designed for transparent and optimized wireless communication. This module is flexible to use as both master or slave configuration. Once the module is interfaced with any master controller such as computer, smart phones it starts to transmit data continuously over the air.

➢ **Specifications:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
	Length (mm)	26.9	
	Width (mm)	13	
	Height (mm)	2.2	
	Weight (g)	5	
	Operating voltage (V_{CC})	5 V	
	Operating Current (mA)	30	1

HC-05 Bluetooth Module	Transmission Range (m)	10	
	Bluetooth Version	2.0 + EDR	
	Communication Interface	UART/TTL	
	Operating Frequency	2.4 GHz ISM Band	
	Default Password	1234 or 0000	
	Default Communication	Slave	
	Default Baud rate	9600	
	Default Firmware	LINVOR	

- **Component Diagram:**

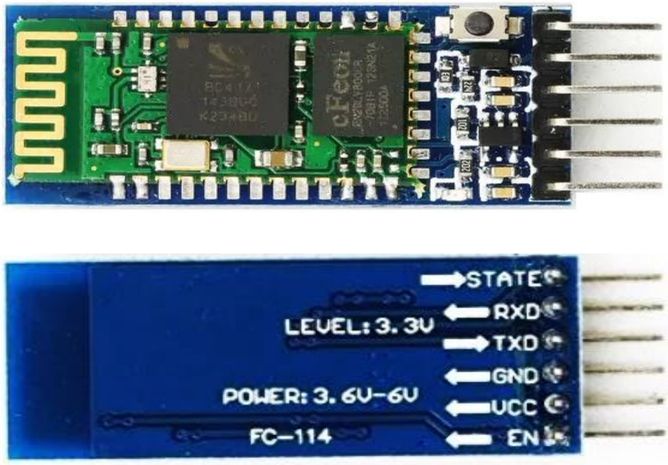


Figure 27: Pin-out Diagram of HC-05 Bluetooth Module

- **Features:**

1. Effortless pairing and fast data transmission.
2. Highly stable and reliable performance.
3. User-friendly and easy to integrate into projects.
4. Supports multiple communication protocols.
5. Low power consumption for energy efficiency.
6. Flexible and compatible with various devices.
7. HC-05 has **red LED** which indicates **connection status**. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.
8. The data transfer rate of HC-05 module can vary up to **1Mbps** is in the **range of 10 meters**.

➢ Interfacing HC-05 with Arduino Uno:

HC-05 Bluetooth Module Pins	Arduino Uno Pins
Vcc	5 V
GND	GND
TX	RX
RX	TX

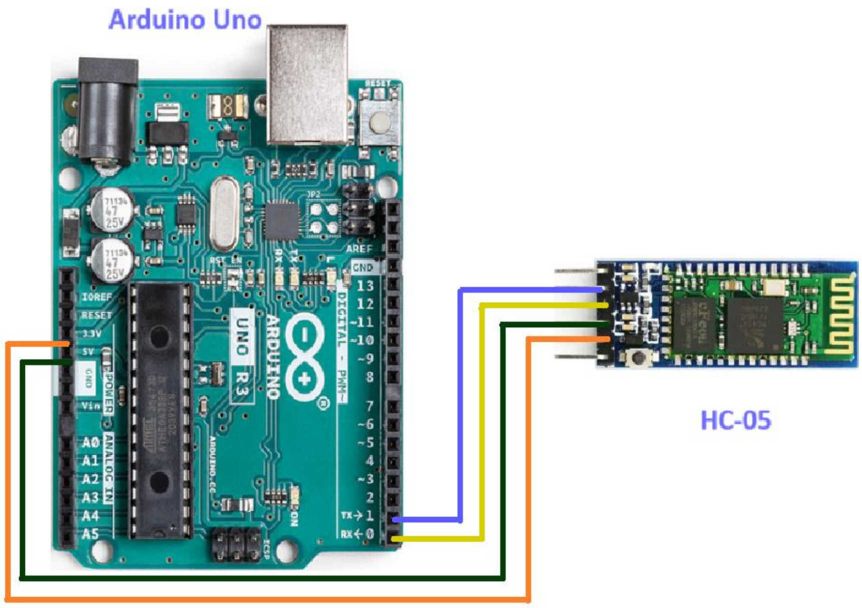


Figure 28: Pin-Out Connection of HC-05

➢ **Applications:** HC-05 Bluetooth Module is being used widely in varieties of applications which are as discussed below.

1. HC-05 Module is used in Internet of Things (IoT) projects.
2. It is also used in Home Automation and Smart Control Systems.
3. When we required wireless data transfer between devices we can use HC-05 Module.
4. Robotics and automation applications has use of HC-05.
5. Wearable technology and health monitoring devices has wide application of HC-05.
6. It is also used in Industrial automation and control systems.
7. Moreover, Audio and sound equipment connectivity uses HC-05 Bluetooth Module.

10.3.3 L298N MOTOR DRIVER:

- **Description:**

The L298N Motor Driver [56] is a popular dual H-bridge motor driver integrated circuit (IC) commonly used in robotics and automation projects. It allows for easy control of DC motors or stepper motors in both forward and reverse directions. Each H-bridge within the L298N can effectively manage the flow of current to its corresponding motor, making it suitable for a wide range of motor control applications. Whether driving DC motors or stepper motors, the L298N's robust design and high current handling capacity ensure reliable performance in diverse scenarios. Additionally, its compatibility with a wide voltage range, typically spanning from 5V to 35V, adds to its versatility, enabling integration with various power sources and motor types. We can refer to the outline diagram of L298N Motor Driver for better understanding.

- **Specifications:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
L298N Motor Driver	Length (mm)	43	2
	Width (mm)	34	
	Height (mm)	27	
	Weight (g)	48	
	Operating voltage (V_{CC})	5 V to 35 V	
	Operating Current (mA)	36	
	Peak Current	2 A	
	Power Consumption (Max)	25 W	
	Operating Temperature	-25 ^oC to 135 ^oC	
	Controlling Level	Low: -0.3V~1.5V high: 2.3V~Vss	
	Enable Signal Level	Low: -0.3V~1.5V high: 2.3V~Vss	
	Number of Channels	2	

➢ Component Diagram:

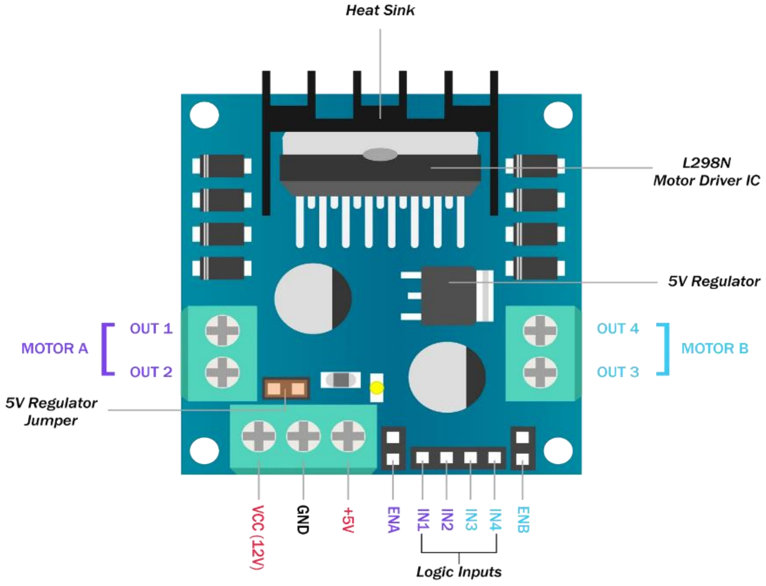


Figure 29: L298N Motor Driver Pin-Out Diagram

➢ Interfacing L298N with Arduino Uno:

Basically, Two L298N Motor drivers has been used in this project. One driver operates the front two motors and the other L298N driver operates the backward motors of the model. The pin-out connection for interfacing these two motor drivers with Arduino uno is as given below:

Front L298N Driver Pins	Arduino Uno Pins
IN 1	6
IN 2	7
IN 3	8
IN 4	9

Back L298N Driver Pins	Arduino Uno Pins
IN 1	3
IN 2	4
IN 3	12
IN 4	11

We can refer to the Chapter 10.2 of Hardware connection diagram for better understanding about the connection.

- **Features:**

1. L298N motor driver can operate at higher voltage and current.
2. It has 4 LED pins that reflects the state of control logic.
3. It provides the features of independent direction, speed and braking for each motor.
4. The driver has a heavy-duty heat sink for maximum performance.

- **Application:**

1. L298N motor driver IC is used in different fields like robotics, embedded, etc. We know that microcontrollers work with very little voltage and current but motors use high voltage and current. So, for this reason, motor driver ICs are used to provide high voltage and current.
2. This motor driver is used in high power-based applications.
3. This IC is used where current control & PWM operable IC is required.
4. This IC is used where the control unit provides only TTL outputs.

10.3.4 DC MOTORS:

- **Description:**

A DC (Direct Current) motor is a fundamental electromechanical device used for converting electrical energy into mechanical motion. It operates on the principle of electromagnetic induction, where the interaction between a magnetic field and current flowing through wire coils generates rotational motion. Basic components of a DC motor include:

1. **Stator**: The stationary part of the motor that houses the magnetic field. It typically consists of permanent magnets or electromagnets.
2. **Rotor:** The rotating part of the motor, also known as the armature, which contains the wire coils. When current flows through the coils, it generates a magnetic field that interacts with the stator's magnetic field, causing the rotor to rotate.
3. **Commutator:** A segmented metal ring attached to the rotor shaft, which reverses the direction of current flow in the wire coils as the rotor spins. This reversal ensures continuous rotation in the same direction.
4. **Brushes:** Conductive contacts that maintain electrical contact with the commutator, allowing current to flow into the wire coils as the rotor rotates.
5. **Shaft:** The output shaft connected to the rotor, which transfers the rotational motion to external components such as wheels or gears.

➢ Specification:

NAME	PARAMETER	SPECIFICATION	QUANTITY
DC Motor	Length (mm)	70	4
	Width (mm)	50	
	Height (mm)	50	
	Weight (g)	200	
	Input Voltage (V)	12	
	RPM	1000	
	Torque (Kg-cm)	5	
	Shaft Length (mm)	25	
	Max. Load Current (mA)	330	

➢ Component Diagram:

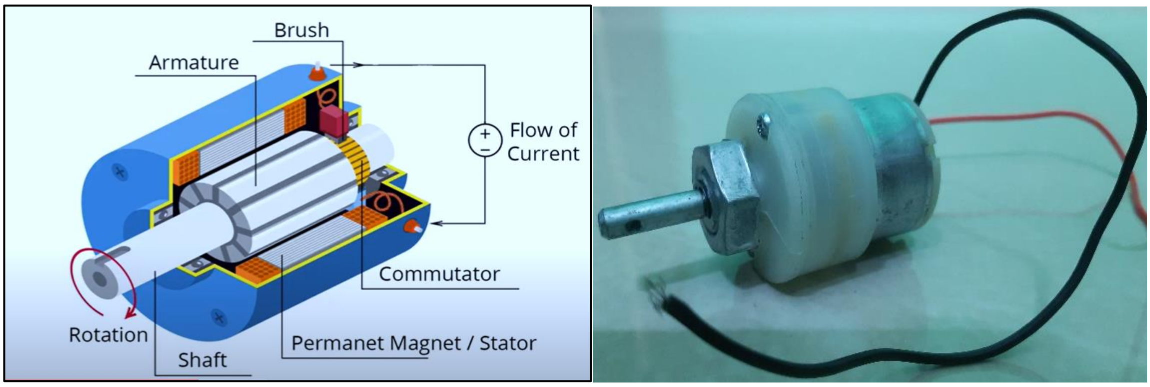


Figure 30: Outline Diagram of Geared DC Motor [57]

➢ Features:

1. DC motors are relatively easy to control using varying voltage levels or pulse-width modulation (PWM), allowing precise speed and direction control.
2. They can provide high torque at low speeds, making them suitable for applications requiring power and torque, such as robotics and vehicles.
3. DC motors are often compact and lightweight, making them ideal for integration into small-scale robotics projects.
4. Compared to other types of motors, DC motors are generally more cost-effective, making them accessible for hobbyists and educational projects.

- **Application:**

DC motors are valued for their reliability, precise control, and versatility, contributing to advancements in technology and automation. It is used in robot locomotion, arms, and grippers due to precise control. It is also popular in model trains, remote-controlled cars, and DIY robotics. Moreover, they are found in shavers, vacuum cleaners, kitchen appliances, power infusion pumps, surgical tools, diagnostic equipment, solar tracking systems and wind turbines.

10.3.5 Li-Po BATTERY:

- **Description:**

Li-Po (Lithium Polymer) batteries are a type of rechargeable battery that have become popular in various electronic devices due to their high energy density, lightweight design, and ability to deliver high discharge rates. LiPo battery uses lithium-ion technology. They offer cost savings for users. These batteries typically have a positive electrode made of Lithium-cobalt oxide **($LiCoO_2$)** or Lithium iron phosphate **($LiFePO_4$)**, and a negative electrode made of carbon. The electrolyte varies depending on the battery type. During discharge, lithium ions move from the anode to the cathode, creating free electrons and generating a charge. When the battery is recharged, the opposite occurs, with lithium ions moving from the cathode to the anode. This process allows for the flow of electrons, powering the device. Here in this prototype we have used two 12 Volt 2200 mAh 80C rechargeable Li-Po batteries for the operation purpose.

- **Specification:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
12 V Li-Po Battery	Length (mm)	110	2
	Width (mm)	34	
	Height (mm)	25	
	Weight (g)	180	
	Capacity (mAh)	2200	
	Output Voltage (V)	11.4	
	Charge Rate (C)	1 to 3	
	Max. Charge Rate (C)	5	
	Discharge Plug	XT-60	

	Balance Plug	JST-XH	
	Discharge	80 C	
	Number of Cells	3	
	Voltage/Cells	3.8 V	

➢ **Component Diagram:**

Figure 31: Outline Diagram of 12V Li-Po Battery [58]

➢ **Features:**

1. LiPo batteries utilize lithium-ion technology, but with a different electrolyte composition. They typically consist of multiple cells connected in series or parallel to achieve the desired voltage and capacity.
2. LiPo batteries offer a higher energy density compared to traditional nickel-based batteries, making them ideal for applications where weight and size are critical factors.
3. LiPo batteries come in various configurations, offering different voltage and capacity options to suit specific requirements. Common voltages include 3.7V per cell, with capacities ranging from a few hundred mAh to several thousand mAh.
4. LiPo batteries are capable of delivering high discharge rates, making them suitable for applications that require bursts of high power, such as RC vehicles, drones, and high-performance electronic gadgets.
5. LiPo batteries can be charged relatively quickly compared to other battery chemistries, provided that they are charged using a compatible charger and charging protocol to ensure safety and longevity.

➤ Application:

Li-Po batteries find extensive applications across various industries and consumer electronics due to their lightweight design, high energy density, and rechargeable nature. Some common applications include: RC vehicle and drones, Power Banks, Laptop, Smartphones, Electric Vehicles, Renewable Energy Storages and many more.

10.3.6 JUMPER WIRES:

➤ Description:

Connecting wires or Jumper wires are used to inter-connect different components. Jumper wires are essential components in electronics and prototyping projects, serving as flexible connectors to establish electrical connections between various components on a breadboard or between different points on a circuit board. They consist of thin, flexible wires with connectors at each end, typically male pins or female sockets, which allow for easy insertion into breadboard holes or connection points on electronic components. Jumper Wires are basically of four types which are: Male-to-Male, Male-to-Female, Female-to-Female and Dupont Jumper wires.

➤ Specification:

NAME	PARAMETER	SPECIFICATION	QUANTITY
Jumper Wires	Length (mm)	200	2
	Conductor Diameter (mm)	0.6	
	Insulation Thickness (mm)	0.25	
	Tensile Strength (Kg/Cm^2)	120	
	Operating Temperature	20 ºC	
	Loop Resistance (Ω/Km)	132	

➤ Component Diagram:

10.3.7 PVC PIPE & 60^0 ELBOW:

➢ **Description:**

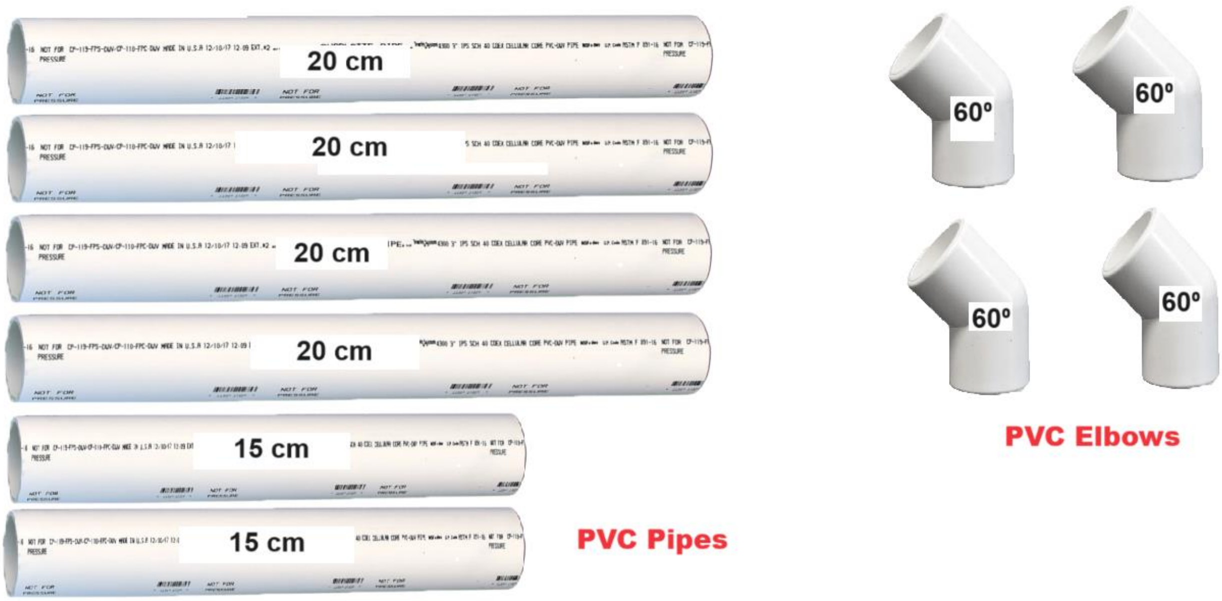
In this project to make the outer structure of the vehicle we have used PVC pipes. We have cut a 7.5 feet long PVC pipe into small pieces as per our requirement. The measurements of each pipe have been shown in the specification section. Moreover, how many pieces of each length is required is demonstrated below. We first cut the pipes in specific lengths and then joint them with the 60^0 PVC elbow to make a car structure. We can refer to the below diagram for better understanding.

➢ **Specification:**

NAME	PARAMETER	SPECIFICATION
PVC Pipe	Length (mm)	190
	Diameter (mm)	18
	Material	PVC
	Colour	White

NAME	LENGTH	QUANTITY
PVC Pipe	20 cm	4
	15 cm	2
PVC Elbow	60^0 Elbow	4

➢ **Component Diagram:**



10.3.8 CHASSIS:

- **Description:**

The chassis of an RC car model serves as the framework or structure that houses and supports all the components of the vehicle. It plays a crucial role in providing stability, durability, and structural integrity to the RC car. Chassis designs can vary widely depending on the type of RC car. Referring to the below figure, in this model we have used a chassis of 31/21 cm made of plywood material. We have also cut four holes at the four corners of the chassis which will help in fitting of the PVC pipes, to finally make the car structure of the model.

- **Specification:**

NAME	PARAMETER	SPECIFICATION
Chassis	Length (mm)	310
	Width (mm)	210
	Height (mm)	10
	Weight (g)	150

- **Component Diagram:**

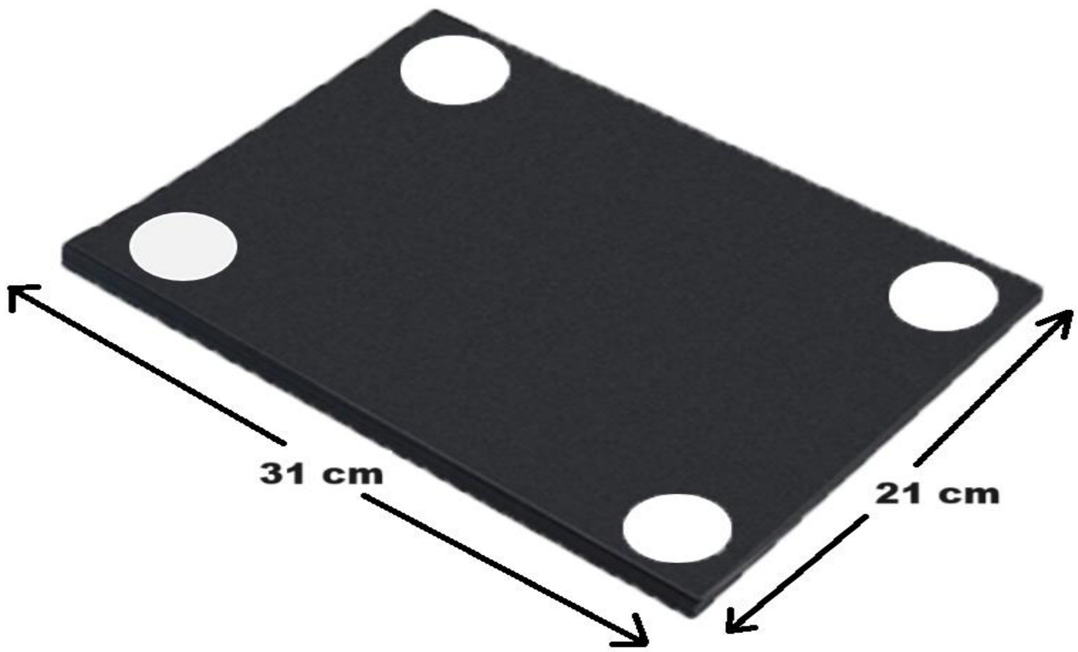


Figure 32: Chassis with Dimensions

10.3.9 HOSE CLIP:

- **Description:**

A hose clip, also known as a hose clamp used in fittings of different things. It typically consists of a circular metal band with a screw mechanism that tightens around the hose, compressing it onto the fitting. The band of the hose clip is usually made of stainless steel or another corrosion-resistant material to withstand exposure to moisture and chemicals. The screw mechanism allows for easy adjustment of the clamp's diameter to accommodate different hose sizes or to achieve the desired level of tightness. Here we have used Hose-Clamp to tighten the DC motor at the end point of four PVC Pipe legs. We can refer to the below figure for better understanding.

- **Specification:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
Hose Clamp	Dimensions (inch)	0.5 to 5	4
	Band Width (mm)	12	
	Weight (g)	250	
	Bolt Size (mm)	7	
	Material	Galvanised Mild Steel	
	Material Grade	202	

- **Component Diagram:**

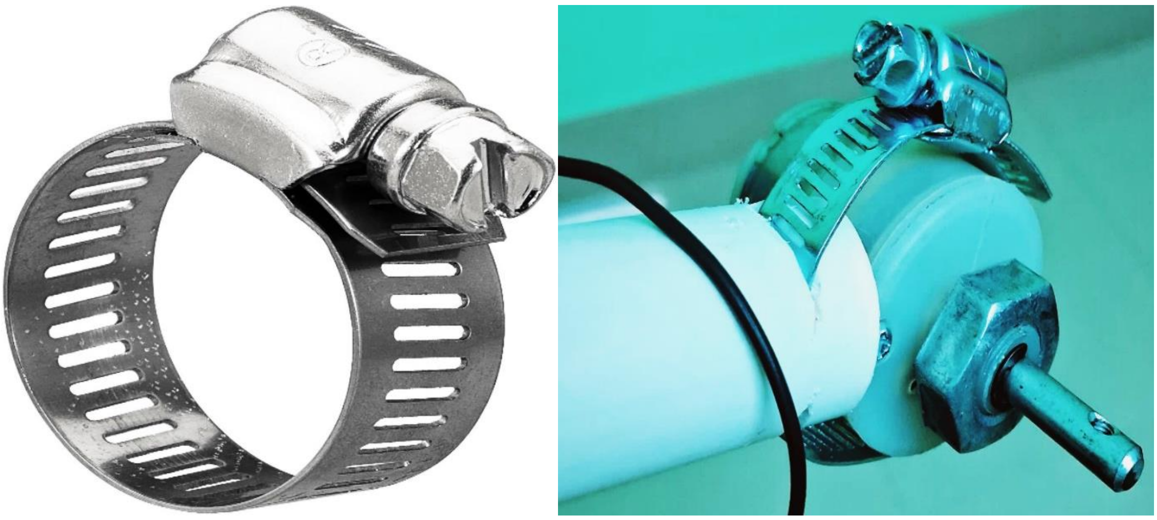


Figure 33: Demonstration of Hose-Clip in Prototype

10.3.10 WHEELS:

➤ **Description:**

Here in this prototype model, we have used four high griping wheels of large diameter. This wheel helps in better stability of the model as well as increases efficiency of the model effectively. These wheels can be easily attached to the geared DC motors with screws and thus provides better mechanical strength to the prototype. They will be very useful in off-roading scenarios where stability becomes a greater issue.

➤ **Specification:**

NAME	PARAMETER	SPECIFICATION	QUANTITY
Wheels	Diameter (mm)	110	4
	Tire section (mm)	20	
	Weight (g)	200	
	No. of Spokes	6	

➤ **Component Diagram:**

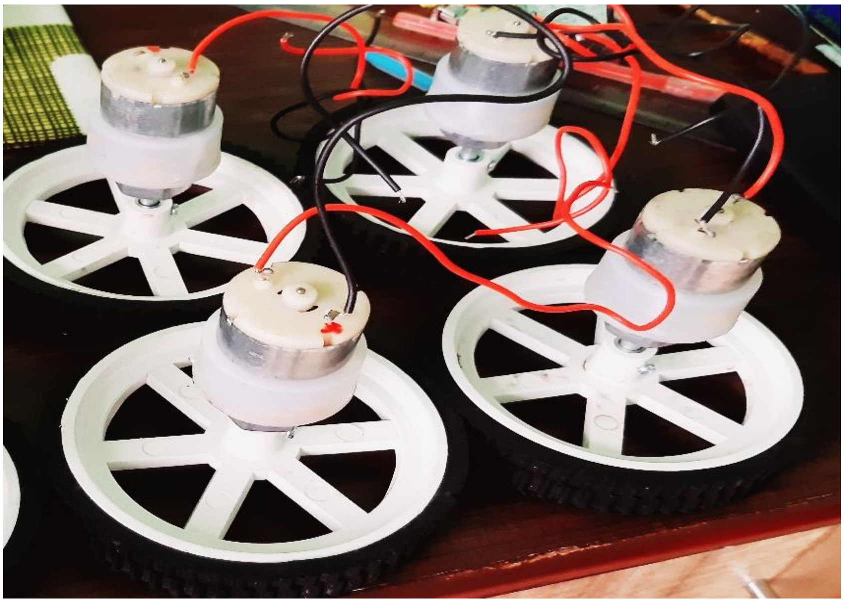


Figure 34: Picture of Wheels Attached with Motors

10.4 METHODOLOGY

In this section, we will discuss about the final implementation stage of the model. So far, we have implemented the sensors integration part, CCT computation part and now we will discuss about the process to implement the driving mechanism part. The process is as written below:

1. At first, we will cut the PVC Pipe into required lengths as shown in the figure earlier.
2. Then we have to fit those PVC pipe pieces. Here the 20 cm pieces will be used as four legs to attach the motor and wheels in the model; whereas the 15 cm pieces will be used as the base of the model. Using 60^0 PVC elbows we will assemble the structure.
3. Now, we will take the chassis and cut four holes at the corners as shown, and then we will put the PVC Elbow through that hole to make it a complete car structure.
4. Now we will cut a small section at the end of four legs (PVC Pipe) such that we can attach Hose-Clip at that point. Now using each hose-clip we will attach four motors followed by four wheels of the vehicle. We can refer to figure 35 for better understanding.
5. After developing the mechanical structure, we will move to the technical section for driving the model. At first, we will connect the Arduino Uno with the HC-05 Bluetooth Module, we can refer to the pin-out connections for interfacing. One thing need to be noted during uploading of code is that we have to open the RX and TX terminal from the Arduino Uno. After uploading the code, we can again reconnect the terminals at required port.
6. Then, we will connect the two L298N Motor driver with the Arduino Uno. The plan is to drive the front motors with one L298N driver and the backward two motors with the other motor driver. For that reason, we will connect the drivers at the desired terminals of the Arduino Uno as shown above.
7. Now, we can connect the Arduin Uno with the computer and then we have to upload the coding required for operation of the vehicle. The code is given in the next section in chapter 10.5
8. After uploading the code, now we have to connect each motor to each driver referring the diagram in chapter 10.2
9. Now we have to connect two 12 Volt Li-Po Batteries to the respective motor drivers and also have to power the Arduino uno with the supply.

10. Now by interfacing the smart phone with the Bluetooth of the vehicle we can operate the car effectively. We can discuss more about the process of controlling the prototype using smart phone in the chapter 10.5 respectively.

10.5 SOFTWARE REQUIREMENTS

There are few software requirements, which are necessary for operating the model. Basically, we need the algorithm to drive the vehicle and also, we need to understand the process of controlling the model. Thus, we have differentiated this section into two parts one is the required algorithm another one is the Controlling of the model using smart devices.

10.5.1 DRIVING MECHANISM CODING:

```
//CAR DRIVING MECHANISM CODE
int in1 = 6;
int in2 = 7;
int in3 = 8;
int in4 = 9;
int in5 = 3;
int in6 = 4;
int in7 = 12;
int in8 = 11;
int data;
void setup()
{
 Serial.begin(9600);
 pinMode(6,OUTPUT);
 pinMode(7,OUTPUT);
 pinMode(8,OUTPUT);
 pinMode(9,OUTPUT);
 pinMode(3,OUTPUT);
 pinMode(4,OUTPUT);
 pinMode(12,OUTPUT);
 pinMode(11,OUTPUT);
}
void loop()
{
  while(Serial.available() > 0)
  {
    data = Serial.read();
    //Serial.println(data);    //Uncomment
this line to print the incomming data
if(data == 'F')
{
  digitalWrite(6,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(11,HIGH);
}
if(data == 'B')
{
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(8,HIGH);
  digitalWrite(9,LOW);
  digitalWrite(12,HIGH);
  digitalWrite(11,LOW);
}
if(data == 'R')
{
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(8,LOW);
  digitalWrite(9,HIGH);
```

```
    digitalWrite(12,LOW);
    digitalWrite(11,HIGH);
  }
  if(data == 'L')
  {
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(3,HIGH);
    digitalWrite(4,LOW);
    digitalWrite(8,HIGH);
    digitalWrite(9,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
  }
  if(data == 'S')
  {
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);
  }
 }
}
```

10.5.2 CONTROLLING OF THE PROTOTYPE:

The controlling of the prototype is differentiated into several steps which are as discussed below;

1. Connecting Smart Phone with Prototype Bluetooth:

At first, we have to install **“Arduino BlueControl”** app from the Google Play store library; in our smart phones.

Figure 35: Google Play Store Location For Application Installing

After installing the application in our smart phones we have to turn on the **Bluetooth** of our device. We have to keep in mind that the prototype is already getting power supply through battery and the HC-05 is in ON state. Then, we have to pair HC-05 with our smart phone and the password will be 1234. Now we have to open the application home screen as shown in figure 38 and then we have to connect the HC-05 Bluetooth module through that application. At the time of interfacing, a dialogue box will show that connected to HC-

05 and then we can proceed to the next stage as the Bluetooth is connected to our smart phones.

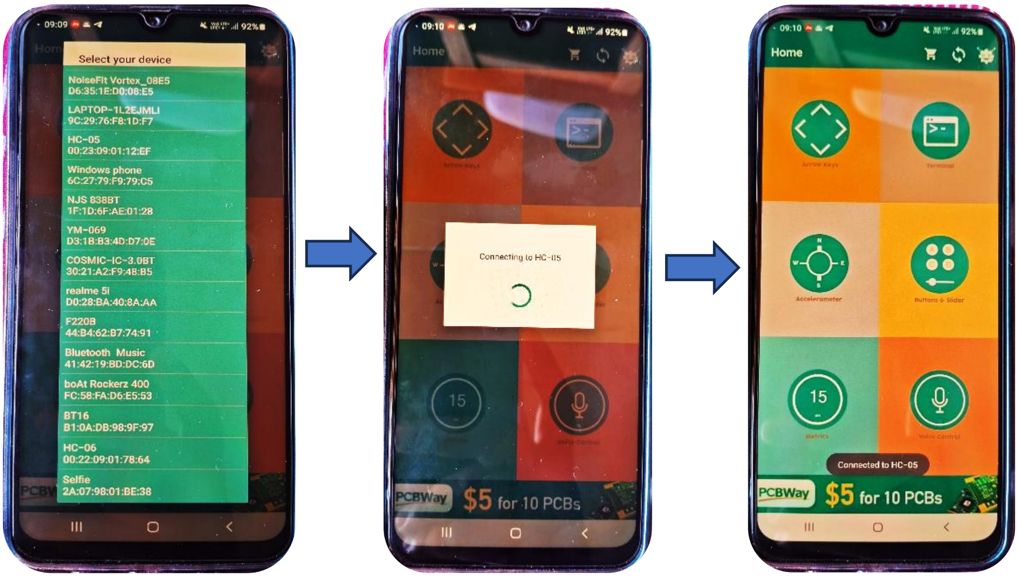


Figure 36: Procedure to connect HC-05 with Smart Phone

2. Controlling The Prototype:

In the next stage, we have to refer to the home screen of the application as shown in figure 39. In the home screen we can see an option "**Buttons & Slider**", we have to click on that option.

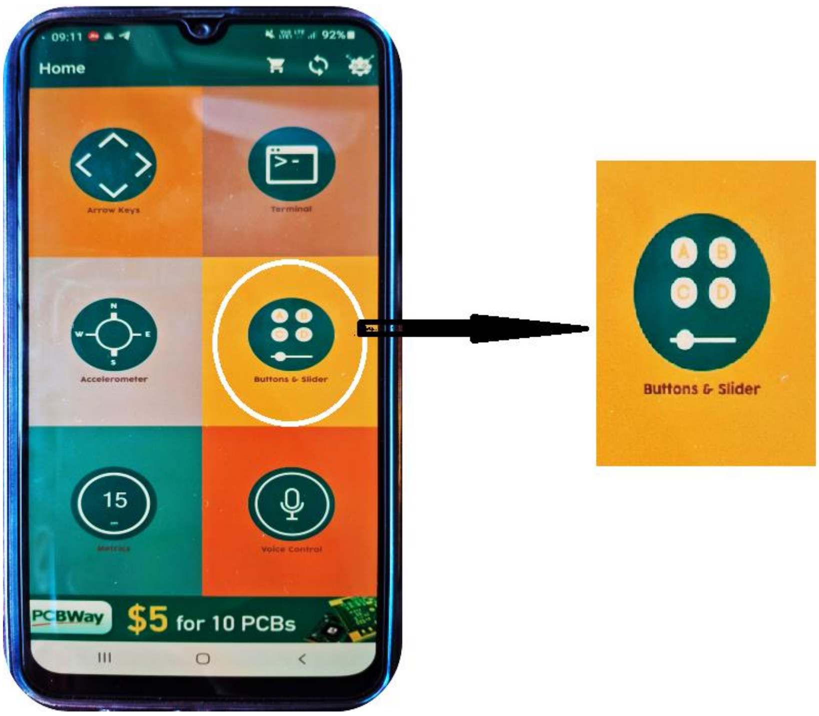


Figure 37: Home Screen of The Application

Now, after opening the button and slider part we have to control the vehicle through some specific commands. We can refer to figure 40 for our better understanding.

Figure 38: Smart Phone Controlling Panel For Prototype Driving

In this screen shown above, there are total six command of which 5 commands (F, B, R, L, S) has been used for controlling the model. There are significant meanings of these commands based on which controlling signal is being send to the prototype model. The commands functions are as discuss below:

- **Command " F " :** Using "F" command we can drive the prototype model in **Forward** direction.
- **Command " B " :** Using "B" command we can move the model in **Backward** direction.
- **Command " R " :** Using "R" command we can move the model in **Rightward** direction.
- **Command " L " :** Using "L" command we can move the prototype model in **Left** direction.
- **Command " S " :** So far, all the above commands are used for driving the model, but at any instant if we need to **Stop** the model, we have to use the command "S". It will immediately stop the car till further input command.

Thus we can obtain a smooth and precise control over the prototype model using smart phone.

CHAPTER 11: OVERVIEW AND VALIDATION OF THE INTEGRATED MODEL

This chapter describes the overview of the prototype hardware scientific model after integrating all the functions as illustrated above. In a quick view to the model from the given below picture we can see that it's an electronic car model which is commonly known as RC Car Model where we have used Bluetooth control for operation and controlling of the car through Smart Phones. Moreover, we can see that the external structure of the model is made up of PVC pipes which provides better flexibility and mobility to the model, at the same time it reduces the overall cost of the prototype. The power source of this model is 12 Volt DC power source where we have used rechargeable LiPo batteries which can be considered as an added benefit of this prototype. There are two compartments in the complete model where in the lower compartment the motor drivers along with the Arduino uno has been placed and in the upper compartment the Sensors has been placed for the measurement purpose. We have to made the model as simple as possible, for better understanding we should refer to the diagrams and pictures of the model.

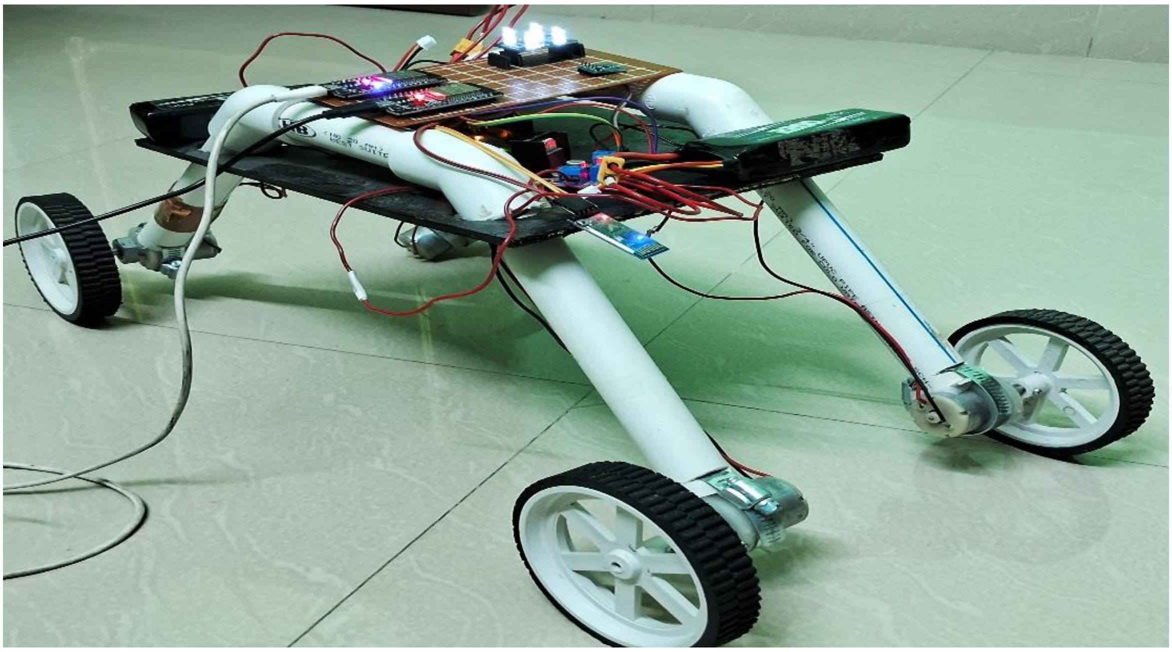


Figure 39: Overview of The Model

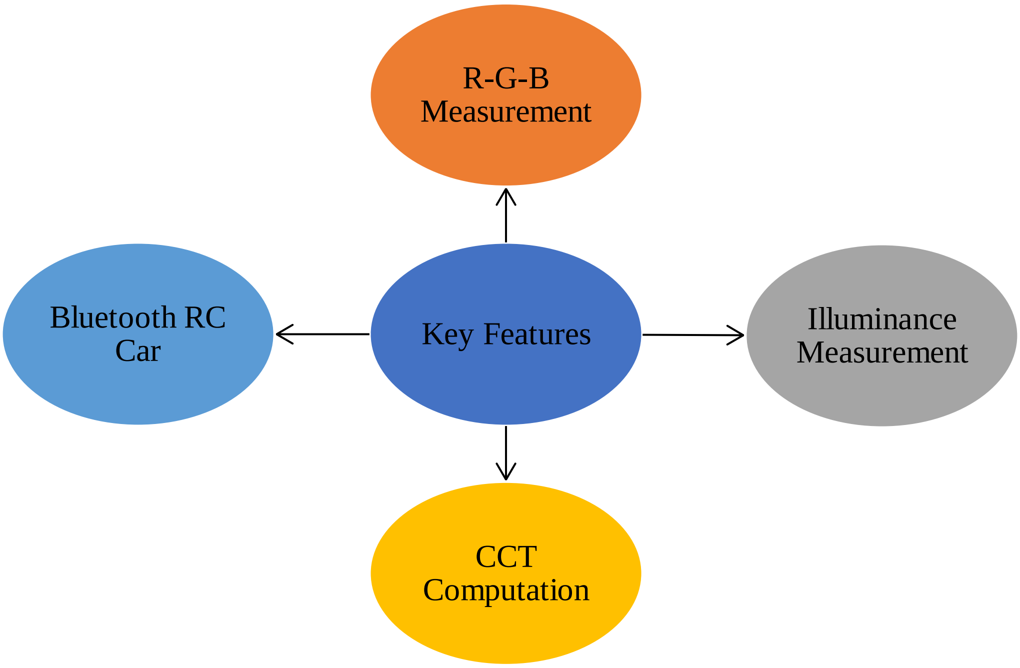
Now, after implementing the integrated model we can now analyse it's features and then we have to validate the model through some practical scenarios. In the next section we will discuss about these points.

11.1MECHANICAL SPECIFICATIONS

NAME	PARAMETER	SPECIFICATION
Prototype Model	**Length**	**61 cm**
	Width	**28 cm**
	Height	**25.4 cm**

11.2 FEATURES OF THE MODEL

The prototype model comprised of features that are essential in the field of photometric measurement. The features are as discussed below:



Considering these features we had elaborated our paper from chapter 8 onwards. R-G-B color content measurement and measurement of the illuminance or lux level are two most important features of this prototype. In photometric measurement these two features are significant and most commonly measured in any lighting design. At the same time CCT computation is necessary for any color tuneable light source not only in interior lighting but also in road lighting scenarios. Thus computation of CCT is also an important feature of this model. Moreover, the driving mechanism of the model is implemented using Bluetooth RC Control through smart phones which provides better flexibility in terms of controlling the model.

11.3 PRACTICAL VALIDATION OF THE MODEL

11.3.1 OBJECTIVE:

To perform an experiment in any standard practical conditions such that the model can be validated. We have to perform this validation for both indoor and outdoor lighting conditions. In the Dark Room Laboratory of Illumination Engineering at Jadavpur University we will perform the validation experiment. We will use a street light set-up and an interior light set-up to perform the experiment. Let's now discuss about each one of them in brief.

Figure 40: Ongoing Experiment For Practical Validation Of The Model

11.3.2 EXPERIMENTAL REQUIREMENTS & DESCRIPTION:

- **Streetlight Experiment:** In this experiment, we will measure the illuminance and calculate the CCT for the Standard SON-T (High Pressure Sodium Vapour Lamp) under standard test conditions. The approach will be to measure the Illuminance and R-G-B values through the sensor set-up of the model. Then we have to calculate the CCT through algorithm. Then, we have to use standard **CL-70 Illuminance Meter** through which, we will measure the illuminance and CCT value at that particular point. And if in the mentioned two procedure, the readings are seeming to be identical. Then we can conclude the model is validated. But for this validation process, some of the standard conditions are: the height of the lighting fixture from Nadir Point is **2.8 m**; the measurement is performed in dark room to prevent inter-reflected components of illuminance. For our better understanding we can refer to the given pictures.

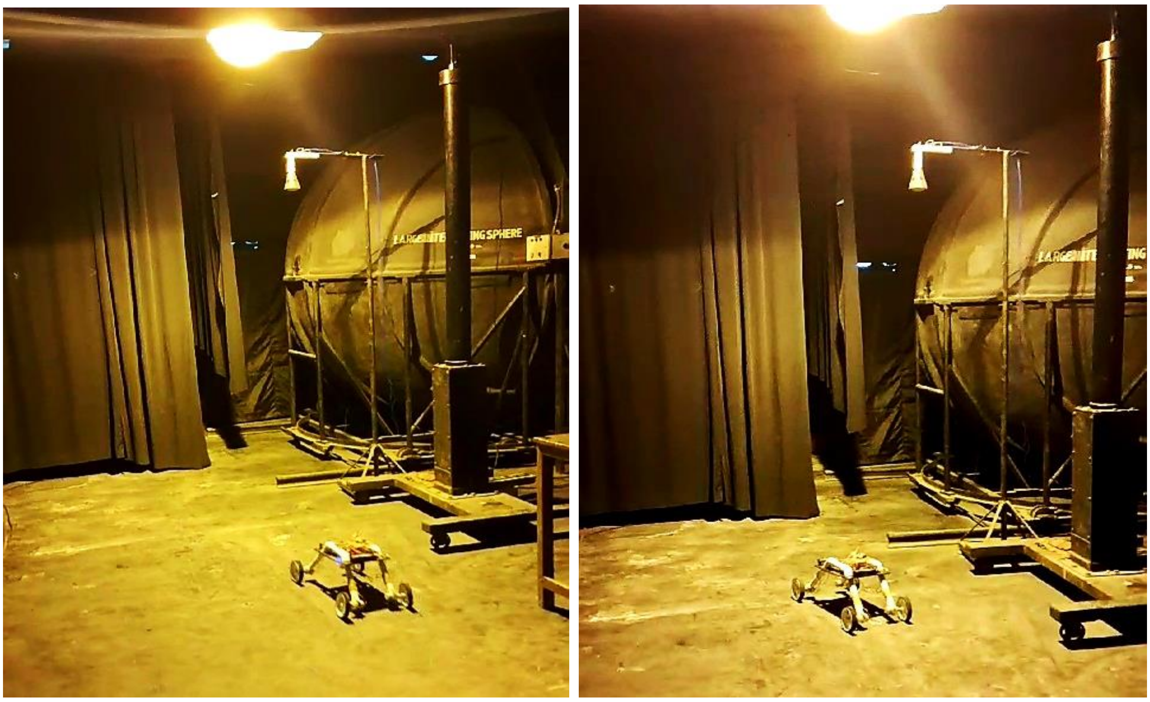


Figure 41: Movement of The Model To the Nadir Point (From Picture Left to Right)

- **Interior Light Experiment:** As our developed prototype model, is useful in both indoor and outdoor applications. Thus, we can measure the Illuminance level and CCT for the interior space as well. To perform experimentation, we have identified a set-up of lighting as shown in figure 41. Using this set-up in the dark room we have placed the model at the centre of the lighting set-up (LEDs) and measured lighting parameters.

Figure 42: Measurement Under Interior Lighting

11.3.3 EXPERIMENTAL RESULT:

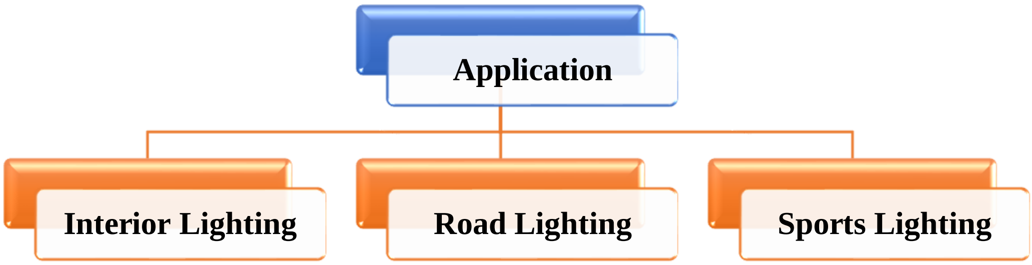
Lighting Type	Height of Luminaire (Meter)	Color Sensor TCS3200 Readings			Calculated CCT (K)	Illuminance Sensor BH1750 Reading (Lux)	Standard Lux Meter Reading (Lux)	Standard CL-70 Color Temperature Reading (K)
		Red (R)	Green (G)	Blue (B)				
Outdoor Lighting (Street Light)	2.8	39	74	87	2517	210	211.5	2450
Indoor Lighting (Interior Light)	3	39	45	50	4500	100	101	4850

11.3.4 RESULT VALIDATION:

By comparing the above readings, we can see that the TCS3200 Color Sensor gives the following R-G-B values through which the calculated CCT is **2517 K** and **4500 K** respectively for both Outdoor and Indoor lighting respectively. As well as, the BH1750 light sensor measures the illuminance level at both the case **210 lux** and **100 lux** respectively. Similarly, the standard Lux meter and the CL-70 Color Temperature measuring device provides almost ideal readings to the practical measurement device such as TCS3200 and BH1750. Thus, we can realize that as the sensor's measured readings are equivalent to the standard equipment results. And this gives the validation of the prototype model in practical scenarios. Now we can conclude that the Prototype model is validated and it is ready for use in practical scenarios such and indoor and outdoor lighting measurement applications.

CHAPTER 12: APPLICATIONS OF THE MODEL

The prototype model has varieties of features as we have seen earlier and that gives this model a wide range of application in the field of lighting. We have distinguished the application of the model based on three broadly classified domain of lighting engineering, which are: **Interior Lighting, Road Lighting** and **Sports Lighting.**



Let's now discuss about the application of the model based on each domain in details:

1. **Interior Lighting:** The above prototype model can be used by Lighting Designers for measuring and analysing the lighting conditions in different areas of a room or building. Using this model we can map illuminance and CCT across a particular interior space. This will help in effective interior lighting design as well as it can optimize the placement and selection of lighting fixture to achieve uniform lighting and desired ambiance. Moreover, it can be used as a lux meter in any working plane of the interior space to measure the average light level. This model can estimate the average light level of any interior space at the same time can estimate the CCT of the particular light source. By giving accurate measurement, this model can increase efficiency and productivity of workers in office, can be useful in human centric lighting and smart home lighting with specific modifications to the model.
2. **Road Lighting:** This prototype model has most significant impact in road lighting design. We can get a real-time data of road lighting design parameters such as illuminance and also, we can determine CCT values of any lighting system. Thus it will be helpful in road lighting design in India as here road lighting design is based on illuminance. **9 Point Method** is one of the most useful methods in determining **Weighted Average Illumiance** of any roadway for a particular lighting fixture. Referring to the method, after marking the specific points **(9 equivalent points)** between two lighting fixtures we can use this model. This prototype can be driven to each point **(P_1, P_2, P_3, P_4, …… P_9)** and at each point we can measure the illuminance level through our prototype. Similarly, using these measurements we can calculate the

weighted avg. illuminance for that particular lighting fixture in the road. We can refer to the below figure for our better understanding about the **9-Point Method.**

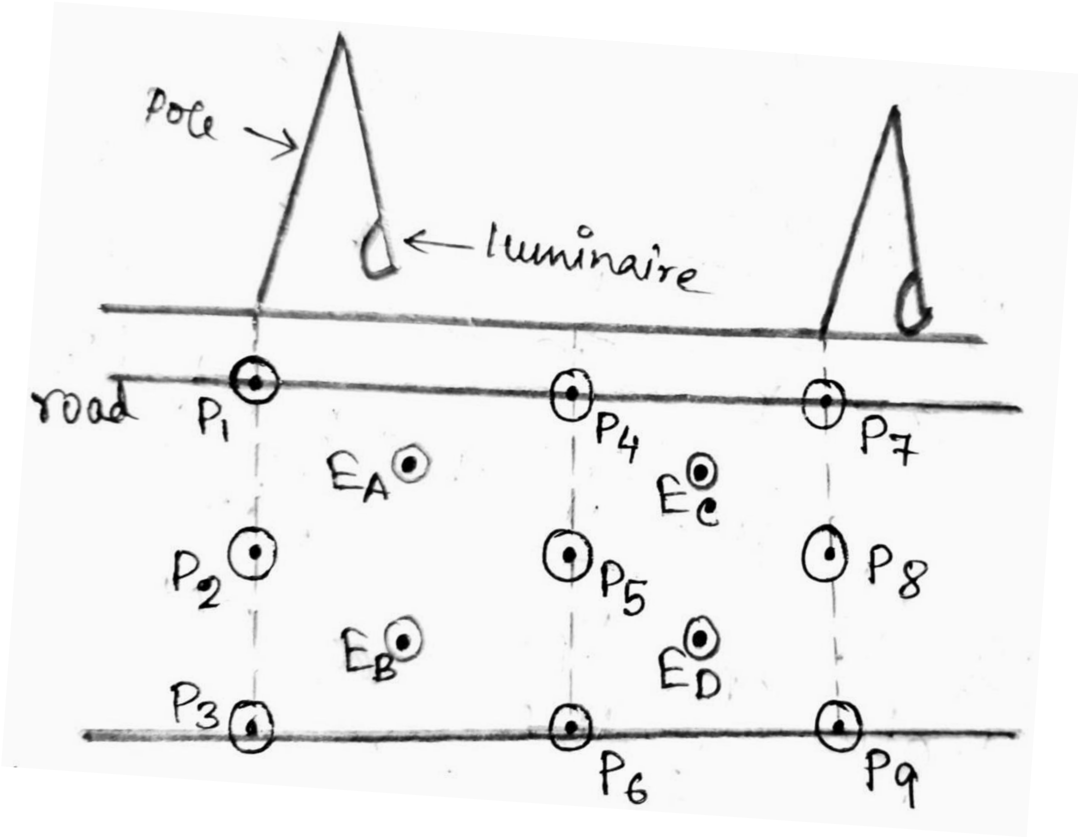


Figure 43: Diagram of 9-Point Method To Determine Avg. Illuminance

Thus, we can use this model in **Road Lighting Design** as well as we can make detailed real time analysis on the road lighting fixture using this model.

3. **Sports Lighting:** Nowadays, the domain of sports lighting has evolved rapidly. From indoor sports to outdoor play, proper lighting conditions are required in each type of sports. Light has significant impact on both mood and aesthetics, and thus also affects human performance. So, accurate lighting design is very important based on televised and non-televised lighting system. And to achieve accurate design, we need to measure the design parameters accurately. Thus, this prototype model can be used to measure average Illuminance of the field are upon which luminaire aiming will be done. So, Horizontal Illuminance as well as CCT of floodlights can be measured using the prototype model.

Thus, this prototype model can be used in different field of lighting industry and so comes the importance of this prototype model.

CHAPTER 13: LIMITATIONS OF THE MODEL

Defining limitations of any scientific model is a continuous process and based on several trial-and-error methods. Thus, any scientific model may come up with different challenges and limitations over the course of time at different challenging circumstances. After evaluating the prototype model several times, the limitations of the model are addressed below:

1. **Sensors Dependency on Calibration:** The accuracy of the measurements heavily depends on the initial calibration of the sensors. Any deviation or error during calibration can lead to incorrect results.
2. **Environmental Factors:** The model's accuracy can be compromised by external light sources or reflections, if ideal conditions are not followed. The performance of sensors may vary with changes in ambient temperature, affecting the consistency of the results.
3. **Measurement Constraints:** The sensors have specific ranges within which they operate effectively. Extremely high or low illuminance levels may not be measured accurately.
4. **Dependency on Look-Up Table:** The look-up table used for CCT computation is based on predefined calibration data, which may not account for all possible variations in light sources.
5. **Scalability:** Scaling the system for larger or more complex environments may require significant modifications and additional calibration, complicating deployment.
6. **Technical Expertise Requirement:** Setting up and maintaining the system requires a certain level of technical expertise, which may not be readily available to all users.

Moreover, the model has certain limitation in terms of movement due to fixed rpm of the motors, and thus during road lighting measurement, if the model need to be operated through ground at a high speed, we might need motors with higher rpm. But though there are few challenges for this model, still there are broad scope of upgradation to overcome certain challenges. In the next section, we will discuss about few upgradation and optimization techniques for making this prototype model a better version of itself.

CHAPTER 14: FUTURE SCOPE OF THE MODEL

Every Scientific Model has certain features that are beneficial for certain working environments, and that also justifies the purpose of the manufactured model. But with time requirement of users' changes, and so we have to continuously make upgradation to any scientific model to make that relevant at that particular timeline. Similarly, this prototype model has certain features that are very significant in lighting industry but to make the model more relevant we need to upgrade the model with some high-end advanced features. Let's discuss about some of the relevant modifications that we can do in coming future.

1. **Integration of Camera:**

- **Purpose:** Make the RC Car autonomous, enhance object detection, and facilitate advanced image processing.
- **Technology:** A high resolution Camera module compatible either with Raspberry Pi or ESP32 can be used in implementing this technology.
- **Impact:** This upgradation will increase the model's relevance for road lighting and other applications requiring visual data.

2. **Automatic Grid Point Detection:**

- **Purpose:** To upgrade remote controlling feature of the car and make the car an automated one, which can solely detect specific grid points for measurement purpose.
- **Technology:** Computer Vision Algorithm which utilizes algorithms such as Harris Corner Detection or Shi-Tomasi Corner Detection to identify distinct points in the visual field. Moreover, through Machine Learning we can train modules using labelled data sets to recognise and predict grid points accurately. By applying Edge Detection Methods (Canny Edge Detector) we can highlights edges of any grid point through image processing.
- **Impact:** This upgradation will reduce the amount of human intervention required for operating the prototype model.

3. **Utilization of Raspberry Pi:**

- **Purpose:** It handles computational tasks and additional processing more efficiently.
- **Impact:** Implementing Raspberry Pi will increase overall performance of the system and moreover it will allow users for more complex computations and real-time data analysis.

4. **Data Storage and Access via ESP32:**

- **Capability:** It can create HTML pages for data visualization.
- **Data Management:** It basically store sensor data at specific IP addresses.
- **Impact:** After implementing this feature, it will improve portability and accessibility of measurement data, enabling remote monitoring and analysis.

5. **Power Consumption Optimization:**

- **Purpose:** In this prototype model we have used Re-Chargeable Li-Po Batteries which is a non-renewable source of energy consumption. Thus, we can use Solar Cells in order to charge those Li-Po batteries.
- **Impact:** Integration of solar cells will make the model more cost effective and eco-friendly at the same time.

6. **Algorithm Optimization for CCT Computation:**

- **Purpose:** Existing algorithm for CCT computation has limitations in accuracy and efficiency if there is a requirement of higher scalability. Thus, we have to optimize the algorithm and code to meet specific requirements.
- **Impact:** This will increase the accuracy and efficiency of CCT measurements, improving overall performance and reliability.

Thus, by following the above methodologies we can modify the model for different requirements and circumstances. These improvements are described in order to provide an idea about the required upgradation to the model. These will increase the model's sustainability, accuracy, reliability and portability. Also enhanced functionality, greater precision and improved efficiency can be achieved through the modifications to the model.

CHAPTER 15: CONCLUSION

The pioneering integration of robotics and photometry through the development of our prototype model marks a significant advancement in the measurement of photometric quantities in real-time. By seamlessly combining RGB and Lux sensors with state-of-the-art algorithms and robotic technologies, our model emerges as a highly versatile tool for a wide array of applications in lighting engineering and beyond.

This research meticulously outlines a comprehensive methodology encompassing sensor calibration, data collection, algorithm development, and model implementation. Through rigorous experimentation and analysis, we have demonstrated the prototype's effectiveness and precision in measuring illuminance, capturing RGB values, and calculating Correlated Color Temperature (CCT). These capabilities not only validate the model's potential but also highlight its practical relevance in real-world scenarios. Despite its impressive capabilities, the prototype model does exhibit certain limitations, such as speed constraints and a reliance on lookup tables for CCT computation. However, rather than viewing these as drawbacks, we see them as exciting opportunities for future enhancements. Our paper proposes several avenues for improvement, including the integration of autonomous navigation, the incorporation of camera-based systems, and the optimization of existing algorithms.

The significance of this research extends far beyond its immediate applications, such as road lighting measurement and interior space analysis. It lays a robust foundation for future advancements in renewable energy integration, efficiency optimization, and eco-friendly design. By addressing current challenges and embracing emerging technologies, this work paves the way for more accurate, efficient, and sustainable photometric measurement solutions. In essence, the integration of robotics and photometry presented in this research heralds a paradigm shift in the field of lighting engineering. It offers innovative solutions to complex challenges and sets the stage for future developments aimed at achieving efficient, effective, and environmentally conscious lighting systems. The journey undertaken in this research not only pushes the boundaries of current technology but also inspires future endeavours in the pursuit of brighter, smarter, and greener lighting solutions.

REFERENCE

1. "COLLATERAL OPTIMISATION CROSS-ASSET AND SUPPORTS ALPHA GENERATION"- David Raccat; ARKETS MEDIA GROUP, 31st August 2022
2. "APPLICATION OF ELECTROMAGNETIC SPECTRUM"- geeksforgeeks, 26th April, 2024
3. "VISIBLE LIGHT COMMUNICATION: DEMAND FACTORS, BENEFITS AND CHALLENGES"- M. Kavehrad, IEEE, 2015
4. "LIGHT MAKES THINGS VISIBLE"- askIITians, April 2023
5. "VIVID SYDNEY LIGHTS UP THE NIGHT"- IDEAS, 11th November 2017
6. "LIGHT, SLEEP AND CIRCADIAN RHYTHMS : TOGETHER AGAIN"- Derk-Jan Dijk, Simon N. Archer; PLOS BIOLOGY, 23rd June 2009
7. "5 DIFFERENT FLUORESCENT TUBE SIZES"- Timothy Thiele, Larry Campbell, 26th October 2023
8. "LEDs FOR SOLID-STATE LIGHTING: PERFORMANCE CHALLENGES AND RECENT ADVANCES"- S. Pimputkar, J.S. Speck, S.P. DenBaars; IEEE Journal, 2009
9. "EVOLUTION OF LIGHT MEASURING INSTRUMENTS: FROM GREASE-SPOT PHOTOMETERS TO ADVANCED SPECTRORADIOMETERS"- J. Smith, A. Johnson, B. Williams; Lighting Research & Technology; Volume: 52; 2020
10. "SEEING THE LIGHT: OPTICS IN NATURE, PHOTOGRAPHY, COLOR, VISION, AND HOLOGRAPHY"- D. S. Falk, D. R. Brill & D. G. Stork; Wiley; 1986
11. "LIGHTING AND THE DESIGN IDEA"- A. Gibson; McGraw-Hill; 2001
12. "THE REPRODUCTION OF COLOUR"- R. W. G. Hunt; Wiley; 2004
13. "INFRARED THERMAL IMAGING: FUNDAMENTALS, RESEARCH AND APPLICATIONS"- M. Vollmer, K. P. Mollmann; Wiley; 2007
14. "THE BLUETOOTH RADIO SYSTEM"- J. C. Haartsen; IEEE; 7(1), 28-36; June 2000
15. "BLUETOOTH TECHNOLOGY: ESSENTIALS AND MOBILE APPLICATIONS"- R. Kraemer & R. J. Bril; Springer Science & Business Media; 2005
16. "DEVELOPMENT OF A REMOTE-CONTROL SYSTEM FOR A MOBILE ROBOT"- J. H. Kim; IEEE; 54(4), 1793-1799; October 2008
17. "INTRODUCTION TO AUTONOMOUS MOBILE ROBOTS"- R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza; MIT Press; November 2011
18. "BH1750FVI: DIGITAL 16-BIT SERIAL OUTPUT TYPE AMBIENT LIGHT SENSOR IC"- Yokogawa Electric Corporation; 2013

19. "TCS3200, TCS3210 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER"- Texas Advanced Optoelectronic Solutions; (2008)

20. "COLOR SCIENCE: CONCEPTS AND METHODS, QUANTITATIVE DATA AND FORMULAE"- G. Wyszecki, W. S. Stiles; Wiley; 2000

21. "COLOR IN BUSINESS, SCIENCE AND INDUSTRY"- D. B. Judd, G. Wyszecki; Wiley; 1975

22. "PROBABILISTIC ROBOTICS"- S. Thrun, W. Burgard, D. Fox; MIT Press; 2005

23. "SPRINGER HANDBOOK OF ROBOTICS"- B. Siciliano, O. Khatib; Springer; 2016

24. "LEARNING OPENCV: COMPUTER VISION WITH THE OPENCV LIBRARY"- G. Bradski, A. Kaehler; O'Reilly Media; 2008

25. "ARTIFICIAL INTELLIGENCE: A MODERN APPROACH"- S. J. Russell, P. Norvig; Pearson; 2021

26. "COMPUTATIONAL PRINCIPLES OF MOBILE ROBOTICS"- G. Dudek, M. Jenkin; Cambridge University Press; 2010

27. "ROBOTICS, VISION AND CONTROL: FUNDAMENTAL ALGORITHMS IN MATLAB®"- P. Corke; Springer; 2011

28. "A STUDY OF CONTROLLING COLOR MIXING OF RED, GREEN, AND BLUE LEDS BASED ON PHOTOMETRY THEORY"- Shinn-Fwu Wang, Hung-Chen Chung, Tsung-Hsun Yang, An-Li Liu, Yuan-Fong Chau, Jeng-Hua Wei, Chiung Chou Liao; American Scientific PublisherS; August 2011

29. "BETWEEN ADDITIVE AND SUBTRACTIVE COLOR MIXINGS: INTERMEDIATE MIXING MODELS"- L. Simonot, M. Hébert; Journal of the Optical Society of America A; 2014

30. "THE ART OF COLOR CO-ORDINATION IN DESIGN"- Perfect Impact; 1st October 2021

31. "TCS3200, TCS3210 PROGRAMMABLE COLOR AND LIGHT TO FREQUENCY CONVERTER"- Texas Advanced Optoelectronic Solutions; Aug 2011

32. "APPLICATION OF THE INVERSE SQUARE LAW IN PHOTOMETRIC STUDIES"- J. Smith, A. Jones; Journal of the Optical Society of America A; August 2018

33. "ANALYSIS OF LIGHT DISTRIBUTION USING THE CUBE LAW IN COMPLEX ENVIRONMENTS"- Y. Wang, D. Lui; Lighting Research & Technology; 2020

34. "COLORIMETRIC AND PHOTOMETRIC PROPERTIES OF A 2° FUNDAMENTAL OBSERVER. COLOR RESEARCH & APPLICATION"- J. Vos; 1978

35. "CIE 1988 2° SPECTRAL LUMINOUS EFFICIENCY FUNCTION FOR PHOTOPIC VISION"- CIE Publication; CIE 1988

36. "LUX MEASUREMENT"- ZLIBRARY; 2016

37. "CL-70F CRI ILLUMINANCE METER"- KONICA MINOLTA

38. "LIGHTING ENGINEERING: APPLIED CALCULATIONS"- R. H. Simons & A. R. Bean; July 2000

39. "ROBOTIC SYSTEM FOR INDOOR ILLUMINANCE MAP GENERATION"- Remy Guyonneau, Franck Mercier, Vincent Boucher; ELSEVIER; 16 February 2024

40. "COLOUR DETECTOR USING ARDUINO UNO AND TCS3200 COLOUR SENSOR"- Tania Teresa Meenattor, Tessa Joise, Ranjitha Rajan; Vol. 04; April 2024

41. "PROTOTYPE OF LOW-COST LUX METER WITH WIDW MEASURING RANGE DESIGNED FOR DYNAMIC LIGHTING SYSTEM"- Roman Hrbac, Tomas Novak, Vaclav Kolar, Mikolaj Bartlomeijczyk; IEEE 2014

42. "COMPARISON BETWEEN LUX METER APPS AND ILLUMINATION MEASURING DEVICES"- Vujica Herzog, N. Dobaj; Chapter 3; DAAAM International Scientific Book 2022

43. "DEVELOPMENT OF WEBCAM BASED LUX METER"- A. H. Ismail, M. S. Muhamad Hashim, M. Nasir Ayob; IEEE 2013

44. "APPLICATION OF THE MECHANISM DESIGN TO DEVELOP THE RGB LEDS COLOR MIXING"- Chih-Ching Hung, Yan-Huei Li, Ping-Han Yang; Hindawi Publishing Corporation; April 2015

45. "PROCESSING RGB COLOR SENSORS FOR MEASURING THE CIRCADIAN STIMULUS OF ARTIFICIAL AND DAYLIGHT LIGHT SOURCES"- Vinh Quang Trinh, Sebastian Babilon, Paul Myland, Tran Quoc Khanh; Applied Science MDPI; January 2022

46. "CALIBRATION METHOD FOR CORRELATED COLOR TEMPERATURE (CCT) MEASUREMENT USING RGB COLOR SENSORS"- Juan-Sebastián Botero, Francisco-Eugenio López; IEEE 2013

47. "SPEED AND DIRECTION CONTROL OF DC MOTOR USING ANDROID MOBILE APPLICATION"- N. Balakrishna, L. Vijay Kumar, S. Sandeep, D. Durga

Prasad, B. Srikanth; EPRA International Journal of Research and Development; Volume 7; June 2022

48. “GERMINATION OF SEEDS IN SMART FARM”- Al. Layla, Noor; ResearchGate; 15th June 2022

49. “BH1750FVI DATASHEET”- Rohm Semiconduction

50. “TCS3200, TCS3210 PROGRAMMABLE COLOR AND LIGHT TO FREQUENCY CONVERTER”- Texas Advanced Optoelectronic Solutions; Aug 2011

51. “INTERFACING TCS3200 COLOR SENSOR WITH ARDUINO”- Last Minute Engineers; 2022

52. “ESP WROOM-32 DATASHEET”- ESPressif System; 2019

53. “ESP-32 PIN OUT DIAGRAM”- Last Minute Engineers; 2022

54. “ARDUINO UNO R3 DATASHEET”- SKU A000066; Arduino Resources; 11th March 2024

55. “DATASHEET BLUETOOTH TO SERIAL PORT MODULE HC-05”- newBluetooth; Electronica60norte; 2022

56. “L298N DATASHEET”- STMicroelectronics

57. “DC MOTOR ARMATURE”- AIP; 26th August 2023

58. “Li-Po BATTERY DATASHEET”- MikroElektronika; 8th March 2013

59. “DESIGN AND IMPLEMENTATION OF ESP32-BASED IOT DEVICES”- H. Al. Shalabi, A. Alsanad; 2021

60. “COMMUNICATION BETWEEN TWO ESP32”- ESP32IO

61. “A COMPREHENSIVE STUDY ON COMMUNICATION BETWEEN ESP32 MODULES USING Wi-Fi”- IEEE Xplore 2019

62. “EFFECT OF COLOR TEMPERATURE AND ILLUMINANCE ON PSYCHOLOGY, PHYSIOLOGY, AND PRODUCTIVITY: AN EXPERIMENTAL STUDY”- K. Choi, S. Jeon, J. Jeong, K. Lee; Energies; MDPI; 2020

63. “THE EFFECT OF CORRELATED COLOR TEMPERATURE AND ILLUMINATION LEVEL OF LED LIGHTING ON VISUAL COMFORT DURING SUSTAINED ATTENTION ACTIVITIES”- S. Yoon, J. Lee, S. Kim, K. Choi; MDPI; 2023

64. “CORRELATED COLOR TEMPERATURE AND LIGHT INTENSITY: COMPLEMENTARY FEATURES IN NON-VISUAL LIGHT FIELD”- M. S. Rea, M. G. Figueiro, J. D. Bullough, A. Bierman; PLOS ONE; 2021