RGB COLOUR PICKER

"Real-Time RGB Color Control Using Arduino and PWM Technology"

Author's name: Kanchan Kumari
Department of Electronics Engineering
SIR M Visvesvaraya Institute of Ttechnology ,
Bengaluru-562157
Email Id: kanchan705032@gmail.com

Author's name: Vaishnavi Vani
Department of Electronics Engineering
SIR M Visvesvaraya Institute of Technology ,
Bengaluru-562157
Email Id: Vanivaishnavi16@gmail.com

***Abstract*—**The RGB colour model is an additive colour model widely used in electronic systems for sensing, representation, and display of images. This project presents an Arduino-based RGB colour picker that enables users to control the colour of RGB LEDs or bulbs remotely via a web interface. The system detects colours from physical objects, reproduces them in real-time on RGB LEDs, and displays their corresponding RGB and hexadecimal values. The system emphasizes accuracy, cost-effectiveness, and user-friendliness, making it ideal for smart home and educational applications.

***Keywords*—**RGB, Arduino, Colour Picker, Internet of Things, LED Control.

I . INTRODUCTION

The RGB colour model is an additive colour model.
The purpose of RGB colour model is used for sensing, representation, and display of images in electronic systems.
ARDUINO is a compact and affordable Wi-Fi microcontroller, ideal for Internet of Things (IOT) projects. One of the engaging applications it can create is an RGB colour picker, allowing users to control the colour of RGB LED strips or bulbs remotely through a web interface.
Easily pick colours from physical objects with this Arduino based RGB colour picker, enabling to recreate the colours to see in real life objects on the pc or mobile phone. Simply push a button to scan the colour of the object and given the RGB colour values as well as an indication of the measured colour on an RGB LED.

II. LITERATURE SURVEY

A. RGB Models and Arduino Colour Pickers

G.Wyszecki., et al(1982) [1] proposed a research work that describes colour science ideas and strategies. The RGB show is used here to acknowledge the shading within the image. The RGB show could be a shading model that joins red, inexperienced associated blue lights in numerous approaches to create an assortment of hues. A colour picker based on Arduino leverages the open-source hardware platform to create a system where users can select and visualize
colours through RGB LED lights or display modules. The colour values are typically controlled through input devices such as potentiometers, buttons, or graphical interfaces. This survey covers the existing research, applications, and challenges in implementing colour pickers using Arduino.

B. Image Segmentation and RGB LED Control with Arduino

R. C. Gonzalez.,et al (1992) [2] in his paper Digital Image process discusses Image segmentation subdivides a picture into its constituent regions or objects. The amount of segmentation depends on the matter to be solved. Non-trivial image segmentation is one among the foremost tough tasks in image process. The accuracy of the segmentation determines the final word success or failure of a computerized analysis program.

Arduino is frequently used to control RGB LED strips, allowing users to pick colours for decorative lighting, ambient lighting in rooms, or stage lighting. These systems often integrate wireless control (e.g., Bluetooth or Wi-Fi modules) for remote colour selection via a mobile app or web interface. The primary research in this domain involves efficient communication between Arduino and other modules (such as Bluetooth, Wi-Fi), and power optimization for controlling large LED arrays.

C. Colour Recognition and Arduino

Abadpour, A.et al (2005) [3] in his paper colour image process using principal element analysis describes the Colour recognition involves comparison of every pixel within the metric and leads to the dominant color because the colour of the given object are explained.

Arduino is a widely-used microcontroller platform in embedded systems, ideal for prototyping electronic projects due to its ease of use, open-source nature, and extensive community support. Arduino boards are often used for creating interactive projects, including colour-picking applications that involve RGB LEDs, displays, and sensors.

RGB LEDs are commonly used in Arduino-based colour picker projects, where users can mix red, green, and blue light to generate a wide range of colours. Arduino controls each colour channel (R, G, and B) through Pulse Width Modulation (PWM) to adjust the brightness of each component.

D. Real-Time Color Recognition and RGB LED Control

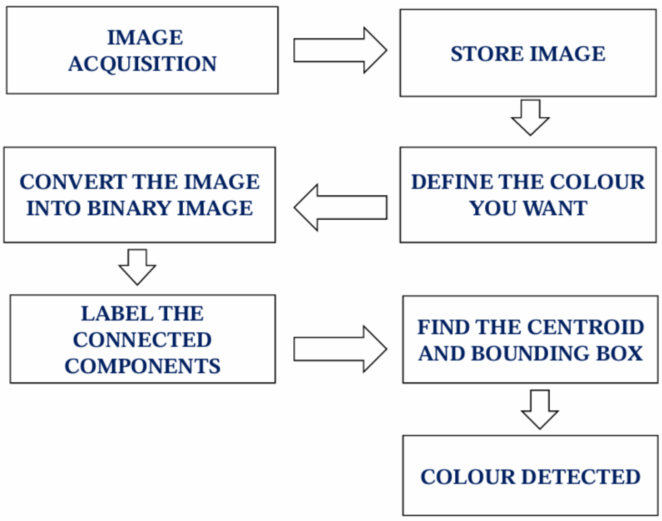
Senthamaraikannan, D.et al (2014) [4] in his paper real time Colour recognition proposes new real time colour recognition features, i.e., extracting primary colours for the aim of vision-based human–computer interaction.

Vision-based human–computer interaction can be achieved by analyzing segmental primary colour regions primarily focused on colour-based image segmentation and vision primarily based colour recognition by addressing these difficulties. However, cluttered backgrounds, unknown lighting conditions and multiple moving objects create this tasks difficult.

Studies on RGB LED control using Arduino emphasize the accurate control of light intensity, power management, and dynamic colour rendering. Research also focuses on the limitations of Arduino's PWM outputs and methods to enhance the control resolution.

III. SYSTEM DESIGN

Fig. 1. BLOCK DIAGRAM:



A. Hardware Details:

- **Microcontroller**: Arduino Board (e.g., Arduino Uno or Nano): Acts as the central processing unit to control the RGB LED and receive input from sliders or potentiometers.
- **RGB LED**: Common Anode or Cathode RGB LED: Allows color mixing by controlling the Red, Green, and Blue components individually. Current-Limiting Resistors (220–330 ohms, typically for each LED channel): Protects the LED from excess current.
- **Potentiometers or Sliders:3** Potentiometers or Analog Sliders: Used to control each RGB channel (Red, Green, Blue) independently, providing input to the Arduino.
- **Alternatives**: Digital input options like rotary encoders or push buttons, but potentiometers are simpler for continuous RGB adjustments.
- **Power Supply**: 5V USB or Battery Pack: Powers the Arduino and LED, ensuring enough current for consistent LED brightness.

Fig. 2. CIRCUIT DIAGRAM :

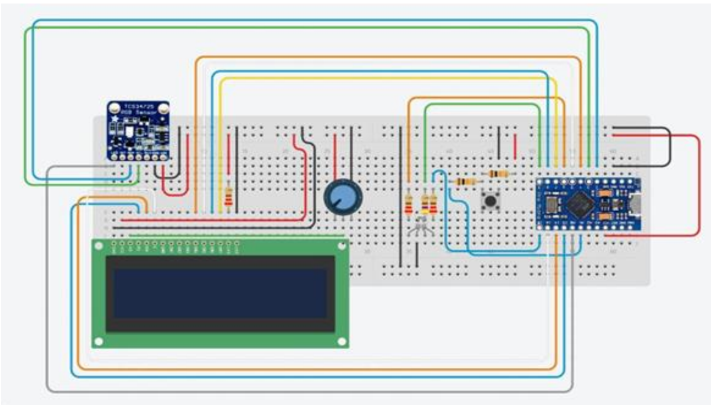
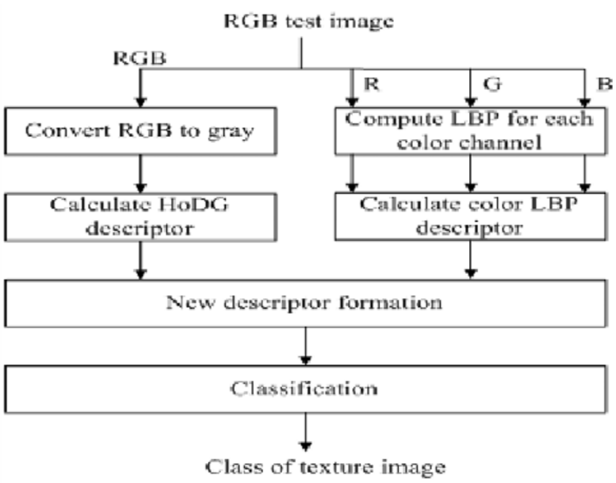


Fig. 3. PROCESS :



B. Software Details:

Arduino IDE : The Arduino Integrated Development Environment (IDE) is a user-friendly platform designed to write, compile, and upload code to Arduino boards. It employs a simplified version of the C++ programming language, making it accessible for beginners while retaining powerful capabilities.

Arduino Code for RGB Control :

Code Structure

*Analog Input Handling***:** Analog values from potentiometers or sliders (range 0–1023) are mapped to RGB values (0–255) for color control.

PWM Output Control: Pulse Width Modulation (PWM) is used to control the brightness of each LED channel (Red, Green, Blue), allowing the creation of a wide spectrum of colors.

Real-Time Update: The RGB LED dynamically updates to reflect the adjustments made to the sliders in real-time.

Display Code

Libraries: For incorporating displays like OLED or LCD, the following libraries are utilized:

- Adafruit_GFX and Adafruit_SSD1306 for OLED displays.
- Liquid Crystal library for standard 16x2 LCDs.

Functionality: Displays the RGB values and hexadecimal codes, which are updated in real-time as users adjust the sliders.

Serial Monitor : The Arduino Serial Monitor serves as a debugging tool, displaying the RGB and hexadecimal values. This functionality is especially useful when an external display is not available.

2.rgb2gary: The loaded image then converted to grayscale which is used in recognition of an object, in analysing of an image and also in image processing.

3.imgaussfilt: Here the Gaussian filter is applied which reduced the noise from the loaded image which may be taken in low light.

4.imfilter: Applying filters like Sobel and Laplacian filter which helps in detecting horizontal and vertical edges of an image by using first-order derivative operators and also helps in highlighting the edges during the continuous changes in intensity basically works on second-order derivative images respectively.

5.imfindcircles: whatever the picture is loaded in the system its work is to detect the circle which is nothing other than iris from the uploaded the image.

6.regionprops: Helps to extract the properties like area and perimeter of the detected iris image.

IV. APPLICATIONS

- Smart homes: Ambient lighting systems.
- Education: Teaching RGB/Hex colour models.
- Art and design: Visualizing colours for creative projects.

V. FUTURE SCOPE

- Integration with IoT for remote control and automation.
- Using AI for improved colour detection and reproduction accuracy.
- Expanding to include more sensors (e.g., light intensity sensors)

VI . CONCLUSION

The system successfully achieved its objectives of providing a reliable and responsive RGB color picker. It demonstrated effective color control, real-time feedback, and integration with displays for visualizing RGB values. The Arduino platform proved to be an ideal choice for this project due to its ease of use, flexibility, and the vast community support that facilitated the development process. This system offers a solid foundation for further improvements, including wireless control and more advanced color recognition features.

VII. REFRENCES

[1] Wyszecki G & Stiles WS, Color science, New York: Wiley, (1982).

[2] Gonzalez RC & woods RE, Digital Image Processing, Addison Wesely Publishing Company, (1992).

[3] Abadpour A, Color Image Processing Using Principal, (2005).

[4] Senthamaraikannan D, Shriram S & William J, Instrumentation Engineering, Vol.2, No.3,(2014).

[5] Balogh, P., & Horváth, G. (2013) RGB LED controlled lighting system (Proceedings Engineering, “Real time colour recognition”,

International Journal of Innovative Research In Electrical, Electronics.).

[6] Atzori, L., Iera, A., & Morabito, G. (2014). The Internet of Things: A survey.

[7] Margolis, M. (2020). Arduino Cookbook. Monk, S. (2020). Electronics Cookbook.

[8] Monk, S. (2017). Programming Arduino: Getting Started with Sketches.

[9] Badamasi, Y. A. (2014). The working principle of an Arduino (in Proceedings of 2014, 11th International Conference on Electronics, Computer and Computation).