Design and Implementation of a Color Mixing Machine

Najmi A Wafi*, Ausama H. Ahmed*, Mahamed A. Saleh*

* Electrical and Electronic, College of Engineering, Engineering, Elmergib University

DOI: 10.29322/IJSRP.14.01.2024.p14534
https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14534

Paper Received Date: 19th December 2023
Paper Acceptance Date: 23rd January 2024
Paper Publication Date: 30th January 2024

Abstract- Control systems have played an important role in the development and growth of modern technology and civilization in recent years. Almost every aspect of our daily lives is affected in some way by some type of control system, which has simplified our lives. One use of automatic control is color matching and mixing, which requires specialized and expensive equipment that can only be utilized in industrial applications and for big quantities. In this paper, an automatic color mixing machine was designed, simulated, and constructed to operate automatically, efficiently, and at a low cost. This machine is controlled by an Arduino Mega microcontroller programmed in the Arduino C language. The machine combines Cyan, Magenta, Yellow, and Key (black) CMYK colors with a white color to create the desired color which is detected by the RGB (Red, Green, and Blue) color sensor. This system is simulated using Proteus software which is a great tool for simulating microcontroller-based systems. Lastly, Python programming language, OpenCV package, and Colour library have been used to calculate Delta E that measures the difference between two colors. The results from this machine was practically acceptable with an error (Delta E) of 12%, 9%, 5%, 8% after testing with four examples (yellow, green, blue, red) respectively.

Index Terms- Color mixing machine, Arduino, Proteus, Peristaltic pump.

I. Introduction

Colors are adjectives for describing things, they have an immense impact on our perception, aesthetic and emotional responses. They help organize our daily life, for instance, traffic light helps avoid accidents. Every color has a wavelength that ranges from 380nm to 750nm. They might be divided into into warm color such as red, orange and yellow, and cool colors such as blue, purple, and green. The process of combining theses colors to get a desired color is called a mixing process. There are a number of models that are categorized into two opposite methods used for mixing colors, additive and subtractive methods. The prior method is achieved by combining light to create new colors. Red, Green, and Blue (RGB) and Cyan, Yellow, Magenta, and Key (CYMK) are considered additive method. The latter method is achieved by subtracting light through the use of light absorbing pigments, Red, Yellow, and Blue (RYB) is an example of this method.

There are people such as artists, who employ the RYB as their primary colors to generate new colors. While people who work with media-related professions such as digital designers work with RGB model. The interaction between primary, secondary (obtained by mixing the primary colors), and tertiary colors (obtained by mixing the primary and the secondary colors) serves as a crucial foundation for the art of color mixing.

Traditional methods of color mixing, such as manual blending or trial-and-error approaches, often involve time-consuming processes and lack the precision necessary to achieve consistent results. Today, an automated color mixing machines either for personal or business use can be found on the market. In this paper a low-cost color mixing machine that uses the subtractive method is proposed.

II. Literature Review

Most works related to an automated color mixing machine are industrial. They depend on using a special model of colors to create another. This is carried out by mixing specific number of colors in different ratios.

A paper by Muftah Mohamed A., published in 2014 presents a process of mixing colors using LABView [1]. The system uses a block-diagram and icons as a control algorithm and RGB color model. The desired color is achieved by mixing the three colors with white color. A prototype of an automated color mixing machine designed for a personal usage was presented in a work titled “Design and development of an automated paint mixing machine”, published in 2014 [2]. The desired color is obtained by mixing the primary RGB color pigments in specific quantities using TCS3414 color sensor and controlled using Arduino ADK board. It is possible to create 12 color with different proportions by a machine built by S. Brindha et. Al. [3]. The colors are produced by mixing RYB colors using motors.The process is controlled using an Arduino with the MATLAB GUI, as the front end, for selecting the desired color. Another system that uses the same RYB color model is presented in 2018 by K. Sanjay et. al. Which uses three containers, one for each color. The system is controlled by Programmable Logic Controller (PLC) and monitored by Human Machine Interface (HMI) screen. Solenoid valves are used to control the flow of the color. A programmable sensor known as EMX color max sensor is used to validate the final generated color. Another work that allows the user to choose up to 12 categories and 12 shades of each color is published in 2022 by Srour A. and Moustafa O. [4]. The system

is controlled by PIC16F877A with a Liquid Crystal Display (LCD) for monitoring. The desired color can be created by mixing RGB and white colors with a minimum amount of 50ml and a maximum of 500ml. The choice of the color is selected by using a keypad.
In this paper, an automatic color mixing machine was designed, simulated, and implemented using an Arduino Mega microcontroller. The system combines RGB color model as an input, using TCS230 color sensor, and CMYK color model for color creation. One peristaltic pump powered by NEMA 17 stepper motor is used for each color.

III. Color Models

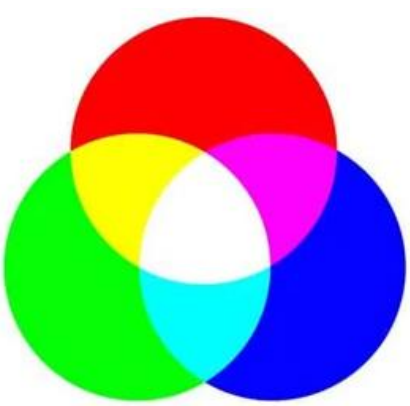
Analyze and understand all the provided review comments thoroughly. Now make the required amendments in your paper. If you are not confident about any review comment, then don't forget to get clarity about that comment. And in some cases there could be chances where your paper receives number of critical remarks. In that cases don't get disheartened and try to improvise the maximum.

Color models, also known as color spaces, are used to define how colors are represented and manipulated. They are essential for tasks such as reproduction, matching and management of colors. Some examples of models are RYB, RGB, CYMK, HSV and hexadecimal color models. In this work, RGB and CMYK are utilized in the proposed system.

A. RGB color model:

This color model is one of the most commonly used color spaces especially for 8-bit digital images. It is used to represent the color for most displays such as monitors, and phone screens. RGB is an additive color model, see Fig. 1, which combines red, green and blue colors in different intensities of the primary colors to produce other colors. The pixels of an image, represented by RGB model, are typically 8-bits with each color provide 256 possible intensities ranging from 0 to 255 or from 0.0 to 1.0 in a floating point systems. If no light is emitted (RGB 0 0 0), black will be perceived, and it is perceived as white when the intensity is at max for all colors (RGB 255 255 255) [5].

Figure 1. RGB color system



B. CMYK color model:

CMYK, shown in Fig. 2, is a four colors model that uses Cyan, Magenta, Yellow and Key (Black) for representation. It is subtractive color model used in printing and photo copying. It represents colors by subtracting the main colors from a white background. CMYK values range from 0% to 100%, for example a white color is perceived if (CYMK 0 0 0 0) is used and black is perceived if (CYMK 0 0 0 100) is used [5].

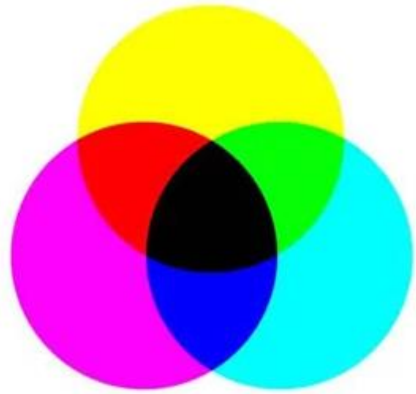


Figure 2. CMYK color system

C. RGB to CMYK conversion algorithm:

An approximation method to convert the RGB color into an equivalent CYMK color is by using the float point system. If the values are in 8-bit format, then they can be convert by dividing the values by 255. Then the CMYK values can be obtained by using the following equations, note that the value of K must be calculated first:

$$K = 1 - max(R, G, B)$$
$$C = (1 - R - K)/(1 - K)$$
$$M = (1 - G - K)/(1 - K)$$
$$Y = (1 - B - K)/(1 - K)$$

where the R'G'B' are the floating point representation of the RGB model.
Finally, the CMYK values can be written in percentage values by multiplying each value by 100 [6], [7].

IV. The Proposed System

The system schematics, depicting the components employed, are presented in Fig. 3. The system is governed by an Arduino Mega board, which starts by utilizing a TCS230 color sensor to acquire the RGB color values from the input sample. Four NEMA 17 stepper motors are driven using A4988 stepper motor drivers, which are connected to the Arduino Mega via a V3 CNC shield. The stepper motors are coupled to four Peristaltic pumps, one pumps for each of the four CYMK colors, responsible for dispensing the required amount of the colors into the mixture.

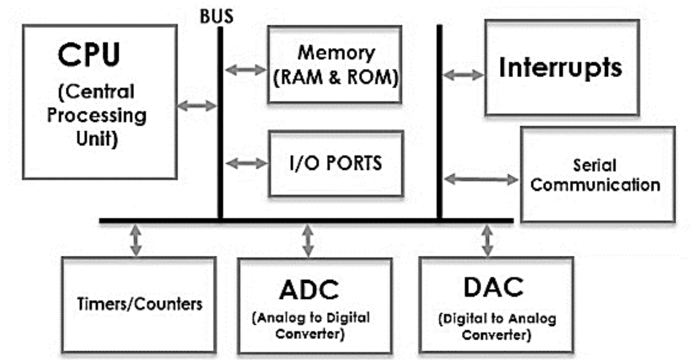


Figure 3. Block diagram of the system

V. System Simulation

This section covers the simulation process of the system components and their operation.

A. Simulating system components

The system is simulated using Proteus Professional software V8.0 as shown in Fig. 4. Some of the real components were not available in the simulation software and replaced with components that operate in similar manner. For instance, three variable resistors were used to input the RGB color values in place of the color sensor. The four Peristaltic pumps were replaced by the stepper motors with LEDs to show their operation.

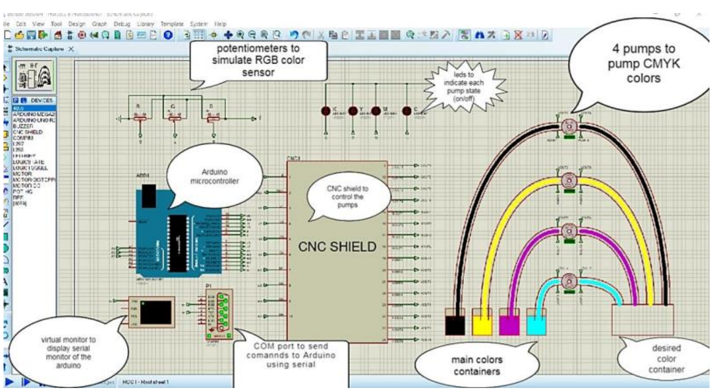


Figure 4. System simulation on Proteus software

B. System operation

The sensor's reading values of the input color, the RGB colors values, are set by changing the values of the variable resistors. This is carried out by moving the slider of each potentiometer left and right as shown in Fig. 5. For example, if we select R=75%, G=24%, and B=78%, which translates to R = 192, G=61 and B=199 values, due to the 1024 resolution of the analog port of the Arduino, will result in readings of R=768, G=244 and B=796. To correct the values to 0 to 255 range, the readings are simply divided by four.

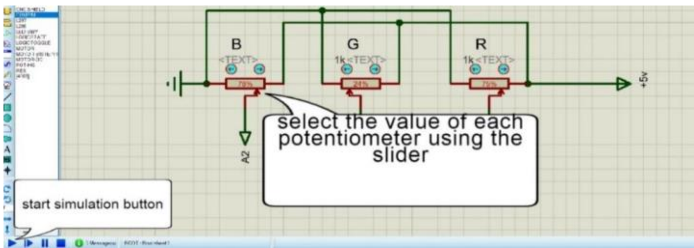


Figure 5. Sensor's readings simulation

The color sensor's readings will be displayed on the virtual terminal. The screen concludes with a prompt that asks the user to enter a command to start the mixing operation. Loading color into the tubes will commence upon executing the "LOAD" command, as illustrated in Fig. 6.

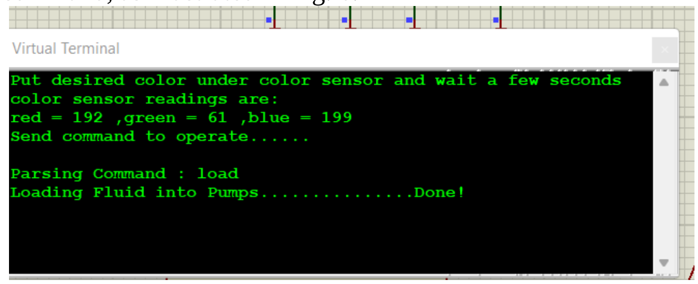


Figure 6. Sensor's readings and load command entering

The desired color is obtained after sending the command (pump r g b ml). The last parameter ml represents the amount of the desired color in ml. The microcontroller will convert the RGB values (192, 61, 199), obtained from the sensor, into the equivalent CMYK (3, 69, 0, 21) and then pours the amount of the paint required. See Fig 7.

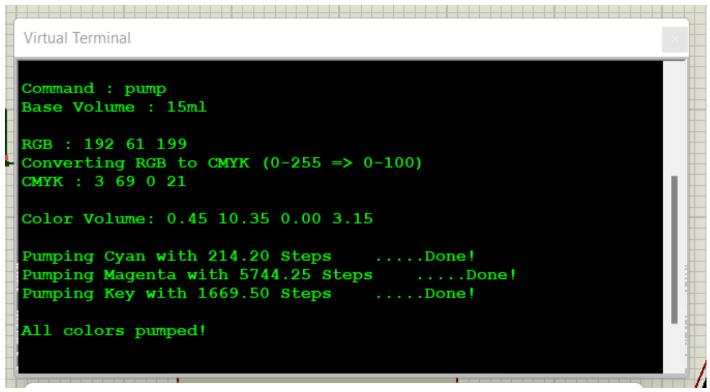


Figure 7. Finish adding colors to the container simulation

The number of steps required for each motor to pump the right amount of color to create the desired result is calculated by the microcontroller. In this case, the steps are (Cyan = 214.2 steps, Magenta = 5744.25 steps, Yellow = 0 steps, Key = 1669.5 steps). In general, the system starts by first reading the RGB values of the desired color using the color sensor. Second, the Arduino converts the reading into CMYK and calculates the quantity of each color to produce the desired color and quantity in ml. Finally, the pumps are operated to convey the right amount of the primary colors into the container.

VI. System Assembly

The components of the system are assembled in a wooden structure, see Fig. 8. The system has an opening in front to put the color sample required, see Fig. 8-a. The four stepper motors, with peristaltic pumps mechanisms, are fixed on the back plate, see Fig. 8-c and d. The Arduino and the rest of the components are placed at the bottom of the machine, see Fig. 8-b.

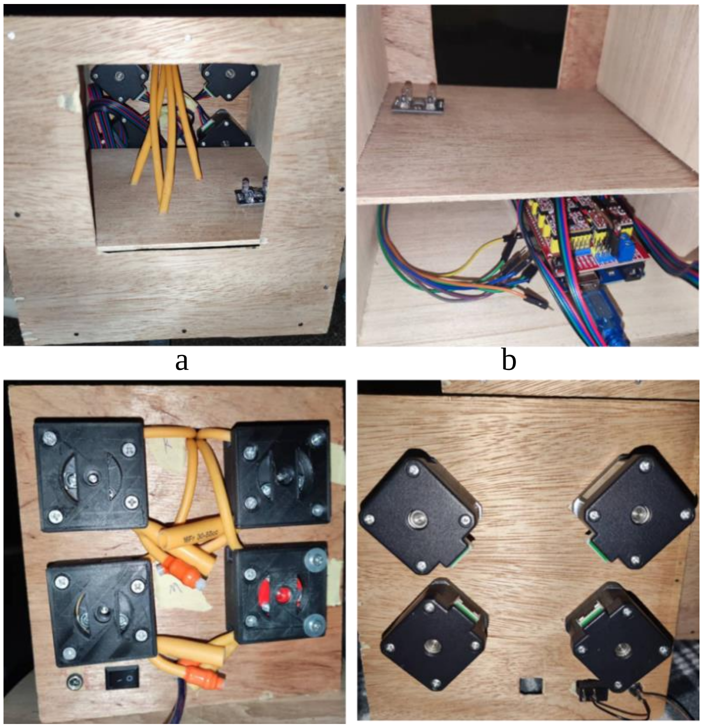


Figure 8. Assembled machine

VII. Results and Discussion

The system was tested by using four color samples and comparing them to the obtained color (output) from the machine, see Fig. 9. Note that the image on the left is the needed color (input), and the one on the right is the produced color (output) in its liquid state. The Delta E metric is used to quantify the precision of produced colors. The range of the Delta E has a range that varies from 0 to 100 and divided into four categories:

- <= 1: Invisible to the human eye
- 1-2: Visible on close inspection
- 2-10: Noticeable at a glance
- 11-49: Colors are more similar than they are different
- 100: Colors are exactly opposite

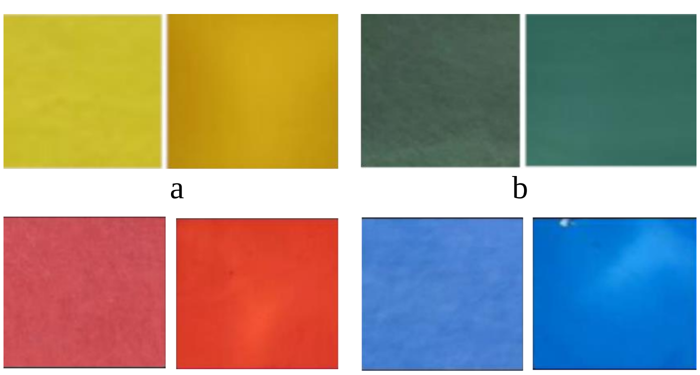


Figure 9. Each subfigure shows input (left) and output (right) colors used to test the machine

The images were compared using Python language with Open-CV, and color library. The results of the comparison is summarized and presented in Table 1. Where the Blue, Red, and Green, are noticeable at a glance, while the yellow is more similar than different.

TABLE 1.

Color	Delta E value	Category of the Delta E
Yellow	12.8057	Colors are more similar than they are different
Green	9.3064	Difference in colors is noticeable at a glance
Blue	5.9897	Difference in colors is noticeable at a glance
Red	8.4594	Difference in colors is noticeable at a glance

VIII. Conclusion

An automatic color mixing machine of was designed simulated and built successfully. It was used to produce four colors (Yellow, Green, Blue and Red) and then test their closeness to the colors used as input. Delta E metric was used to test the similarity of the produced color to the desired one. The system can be improved by using more accurate color sensor. This system is open loop, i.e. there is no mechanism that utilizes the error between the output and the input to correct the produced color; closing the loop would help increase the accuracy of the produced color. Artificial intelligent has the potential to be applied

References

[1] M. A. Muftah, A. M. Albagul, and A. M. Faraj, “Automatic paint mixing process using LabView”.
[2] T. Raj and A. Jena, “DESIGN AND DEVELOPEMENT OF AN AUTOMATED PAINT MIXING MACHINE,” National Institute of Technology, Rourkela India, 2014.
[3] S. Brindha, P. Kishorniya, R. Manickam, K. N. Chakkaravarthy, and C. Poomani, “Automated Color Mixing Machine Using Arduino,” International Journal of Engineering Research & Technology, vol. 6, no. 4, Apr. 2018, doi: 10.17577/IJERTCONV6IS04081.
[4] O. Moustafa and A. Srour, “Automated Color Mixing machine based on Microcontroller,” Islamic University of lebanon, Lebanon, Jan. 2022.
[5] P. U. P. A. Gilbert and W. Haeberli, Physics in the Arts, 1st edition. Amsterdam: Academic Press, 2008.
[6] M. Lindgren and M. Thiel, Automatic Color Mixer : A method for automated color recognition and replication. 2017. Accessed: Jan. 03, 2024. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-226684
[7] V. Chernov, J. Alander, and V. Bochko, “Integer-based accurate conversion between RGB and HSV color spaces,” Computers & Electrical Engineering, vol. 46, pp. 328–337, Aug. 2015, doi: 10.1016/j.compeleceng.2015.08.005.

AUTHORS

First Author – Najmi A. Wafi, MSc, College of Engineering, Elmergib University, nawafi@elmergib.edu.ly.
Second Author – Ausama H. Ahmed, PhD, College of Engineering, Elmergib University, ahahmed@elmergib.edu.ly.

Third Author – Mahamed A. Saleh, Bsc, College of Engineering, Elmergib University, malgaroshy@gmail.com.

Correspondence Author — Najmi A. Wafi, , nawafi@elmergib.edu.ly, +218-924681494.