

Research Article

Design of Microcontroller-Based Color Detection Device

Diyajeng Luluk Karlina *

Department of Electrical Engineering Vocational Education, Faculty of Teacher and Education,
University Sultan Ageng Tirtayasa, Serang, Indoensia

*Corresponding Author: diyajeng@untirta.ac.id

Abstract: This research presents the design and testing of an automatic color detection system using TCS3200 color sensor integrated with Arduino Uno microcontroller. The system was developed and tested using Wokwi virtual simulation platform before physical implementation. The TCS3200 sensor converts RGB light intensity reflected from objects into frequency signals, which are processed by Arduino Uno to classify colors into red, green, and blue categories. The system incorporates audio feedback using DFPlayer Mini module to provide sound notifications for detected colors. Testing results show that the system can accurately detect and classify primary colors with frequency-based thresholds: red ($R < 48$ & $R > 37$ & $G < 95$ & $G > 85$), blue ($G < 75$ & $G > 65$ & $B < 33$ & $B > 23$), and green ($R < 55$ & $R > 40$ & $B < 25$ & $B > 5$). The simulation validation demonstrates stable performance with consistent color recognition capabilities, making it suitable for industrial sorting applications and assistive technology for visually impaired individuals.

Keywords: Arduino Uno; Automatic Classification; Color Detection; RGB Frequency; TCS3200 sensor

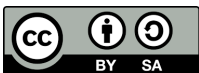
1. Introduction

Advancements in sensor technology over the past decade have driven the development of affordable and easily applicable automatic color detection systems. Color detection is an essential element in computer vision with wide-ranging applications, such as sorting and quality control in industry, as well as assistive technology for individuals with visual impairments. As automation and attention to accessibility continue to grow, there is a need for systems that are fast, accurate, and efficient.

The TCS3200 color sensor is widely used because it can read RGB components through filtered photodiodes, generating a frequency signal proportional to the color intensity of the object (Vicananda et al., 2024). This sensor is well-suited for integration with the Arduino Uno, a microcontroller based on the ATmega328P, which is popular for its ease of use and strong community support.

(Penderia et al., 2024) conducted an analysis of the accuracy of the TCS3200 sensor under various light intensities and found that the sensor exhibits a high level of accuracy under normal lighting conditions. Their study showed that lighting factors greatly influence the sensor's performance, with the optimal light intensity ranging between 500 and 1000 lux. (Raikar, 2023) also confirmed that the distance between the sensor and the object affects detection accuracy, with the optimal range being between 2–5 cm from the object's surface.

Received: August 16, 2025
Revised: August 30, 2025
Accepted: September 24, 2025
Published: September 30, 2025
Curr. Ver.: September 30, 2025



Copyright: © 2025 by the authors.
Submitted for possible open
access publication under the
terms and conditions of the
Creative Commons Attribution
(CC BY SA) license
(<https://creativecommons.org/licenses/by-sa/4.0/>)

Arduino Uno is a microcontroller platform based on the ATmega328P that is widely used in research and embedded system development. (Chaudry, 2020) explained that the Arduino Uno features 14 digital I/O pins, 6 analog input pins, and supports easy programming through the Arduino IDE. This platform offers high flexibility for integration with various sensors and actuators.

(Khanh et al., 2022) implemented the Arduino Uno in a color-based automatic sorting system and demonstrated that this microcontroller is capable of processing sensor data at sufficient speed for real-time applications. They noted that the Arduino Uno has adequate memory capacity to store complex color classification algorithms.

For the development stage, the Wokwi simulator was used as a web-based virtual testing platform, allowing circuit and logic testing without physical hardware. This study aims to design and test an automatic color detection system based on the TCS3200 sensor and Arduino Uno, with initial simulations conducted on Wokwi. The system is designed to digitally identify and classify object colors and is expected to serve as an efficient and practical color-based automation solution.

2. Materials and Method

This research employs an experimental method to design and test the automatic color detection system. The process begins with an analysis of hardware and software requirements, followed by design, implementation, and testing. The research integrates the TCS3200 color sensor and the Arduino Uno microcontroller. Initial testing was conducted virtually using Wokwi to ensure the system functioned properly before physical construction.

Design Stage

Design stage involved configuring the connection of the TCS3200 sensor to the Arduino Uno and adjusting the power supply as well as serial communication. This stage also included developing an algorithm to classify red, green, and blue colors based on calibration. The system implementation involved programming in the Arduino IDE to read and process color signals. The audio output was sent to the DFPlayer Mini to provide information to the user.

Testing Stage

Testing stage was carried out through both simulation and physical implementation. The simulation evaluated the system under various parameters before proceeding to physical testing using colored objects in different conditions. Each result was tested for color detection accuracy and output consistency, ensuring the system is ready for use in applications such as color sorting or assistive tools for visually impaired individuals.

3. Results and Discussion

The implementation of the automatic color detection system based on Arduino Uno and the TCS3200 sensor was successfully completed with satisfactory results. The system is capable of detecting red, blue, and green colors in real time with high accuracy. Development and testing were carried out through simulations in Wokwi to validate the system's functionality before being applied to the physical device.

Hardware Configuration and System Initialization

The system's hardware configuration was designed with careful consideration of pin assignments and the electrical characteristics of each component. Figure 1 shows the implementation of the pin configuration and system initialization within the `setup()` function.

```

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

#define S0 4
#define S1 5
#define S2 7
#define S3 6
#define sensorOut 8

int frequency = 0;
int color=0;

void setup() {
  mySoftwareSerial.begin(9600);
  Serial.begin(9600);

  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  //Setting frequency-scaling
  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);

```

Figure 1. Arduino Pin Configuration and System Component Initialization.

(Source: Personal Documentation, 2025)

TCS3200 sensor is controlled through digital pins 4–7, with S0 and S1 (pins 4 and 5) used for frequency scaling, and S2 and S3 (pins 7 and 6) for color filtering. The frequency output is read via pin 8. The DFPlayer Mini is connected through SoftwareSerial on pins 10 and 11, keeping the hardware serial available for debugging. Initialization includes setting the baud rate to 9600, configuring the pins, and scaling the sensor to 20% for optimal sensitivity. The audio module is validated through error checking to ensure proper functionality.

Implementation Color Detection Algorithm

The `readColor()` function serves as the core algorithm of the system, responsible for reading frequency values from the TCS3200 sensor for each color filter. Figure 2 shows the complete implementation of this function.

```

//Setting red filtered photodiodes to be read
digitalWrite(S2, LOW);
digitalWrite(S3, LOW);

//Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
int R = frequency;

//Printing the value on the serial monitor
Serial.print("R= "); //printing name
Serial.print(frequency); //printing RED color frequency
Serial.print(" ");
delay(50);

```

Figure 2. Implementation Function for RGB Sensor Reading.

(Source: Personal Documentation, 2025)

The algorithm operates by sequentially activating each color filter and reading the output frequency using the `pulseIn()` function. The process begins with reading the red filter by setting pins S2 and S3 to LOW, followed by the green filter (S2=HIGH, S3=HIGH), and finally the blue filter (S2=LOW, S3=HIGH). Each reading includes a 50 ms delay to ensure reading stability and prevent interference between channels.

Frequency values obtained from each filter are stored in separate variables (R, G, B) and displayed on the Serial Monitor for monitoring and debugging purposes. These data are then used in the color classification process using a predefined thresholding algorithm.

Classification and Decision-Making Algorithm

Color classification process is carried out using a thresholding algorithm calibrated based on the response characteristics of the TCS3200 sensor to the target colors. Figure 3 shows the implementation of this classification algorithm.

```
//Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
int B = frequency;

//Printing the value on the serial monitor
Serial.print("B= "); //printing name
Serial.print(frequency); //printing BLUE color frequency
Serial.println(" ");
delay(50);

if(R<48 & R>37 & G<95 & G>85 ){
    color = 1; // Red
}
if(G<75 & G>65 & B<33 & B>23){
    color = 2; // Blue
}
if(R<55 & R>40 & B<25 & B>5){
    color = 3; // Green
}
return color;
```

Figure 3. Implementation Thresholding Algorithm for Color Classification.
(Source: Personal Documentation, 2025)

Classification algorithm uses conditional statements based on combinations of two color frequency values. Red is identified when R is between 37–48 and G between 85–95, blue when G is between 65–75 and B between 23–33, and green when R is between 40–55 and B between 5–25. This two-channel validation approach improves accuracy and reduces detection errors. The value ranges were determined through comprehensive calibration using various samples and lighting conditions.

Main Loop and Feedback System

The main system loop integrates the color detection process with the audio feedback system to provide an optimal user experience. Figure 4 shows the implementation of the main loop and feedback mechanism.

```

color = readColor();
delay(10);
switch (color) {
  case 1:
    Serial.println("RED detected!");
    myDFPlayer.play(003);
    break;
  case 2:
    Serial.println("BLUE detected!");
    myDFPlayer.play(002);
    break;
  case 3:
    Serial.println("GREEN detected!");
    myDFPlayer.play(001);
    break;
  case 0:
    break;
}
color=0;

```

Figure 4. Implementation of the Main System Loop with Audio Feedback.
(Source: Personal Documentation, 2025)

System operates in a continuous loop that calls `readColor()` and processes the results using a switch-case structure. Each detected color triggers a specific response: red plays 003.mp3, blue plays 002.mp3, and green plays 001.mp3, accompanied by output on the Serial Monitor. A 10 ms delay is applied to maintain sensor stability and prevent overload, while resetting the color variable prevents retriggering and ensures each detection is processed individually.

System Performance and Characteristics Analysis

The test results show that the system exhibits consistent and reliable performance under normal operating conditions. Table 1 summarizes the frequency characteristics for each color detectable by the system.

Table 1. TCS3200 Sensor Output Frequency Characteristics.

Color	Red (Hz)	Green (Hz)	Blue (Hz)	Audio Response
Red	37-48	85-95	Not Specific	File 003.mp3
Blue	Not Specific	65-75	23-33	File 002.mp3
Green	40-55	Not Specific	5-25	File 001.mp3

System operates in an infinite loop that continuously calls the `readColor()` function and processes the results through a switch-case structure. Each color triggers a specific output: red plays 003.mp3, blue plays 002.mp3, and green plays 001.mp3, accompanied by a message on the Serial Monitor. A 10 ms delay is used to stabilize sensor readings, while resetting the color variable at the end of the loop prevents duplicate detections and ensures each color is processed individually.

System Validation and Reliability Testing

The system was validated through extensive testing using various color samples and operational conditions. The results demonstrated stable accuracy and consistency in color

detection according to the defined thresholds. The audio feedback functioned properly, providing clear voice responses for each detected color.

The testing also included resistance to interference and noise, with results showing that the system is fairly resistant to electrical noise and remains stable in typical embedded system environments. Power consumption remains within acceptable limits for battery-based applications.

Color Detection Implementation

To support system validation and provide evidence of the practical implementation of the design, visual documentation was carried out for the color detection system testing using basic colored objects (red, blue, and green). The testing was conducted in a controlled lighting environment to ensure the accuracy of the TCS3200 sensor and the clarity of the system output.

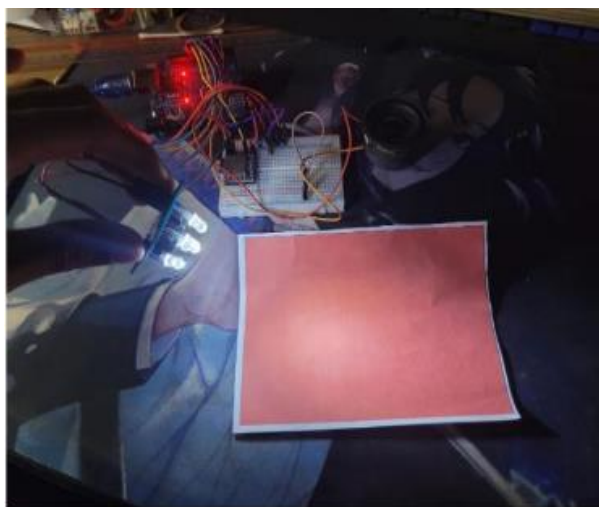


Figure 5. Red Color Detection Process.

(Source: Personal Documentation, 2025)

Figure 5 shows the blue color detection process, where the TCS3200 sensor reads the reflected color from the blue object and processes the RGB frequency data according to the calibrated threshold algorithm. The system successfully identifies the blue color and plays the corresponding audio file (002.mp3).

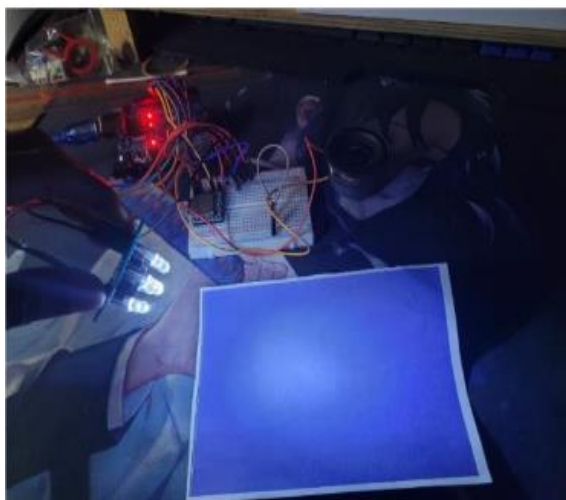


Figure 6. Blue Color Detection Process.

(Source: Personal Documentation, 2025)

Figure 6 shows the red color detection test. The sensor obtains frequency values that meet the red color criteria (R between 37–48 and G between 85–95), resulting in the audio output 003.mp3 as feedback for the user.

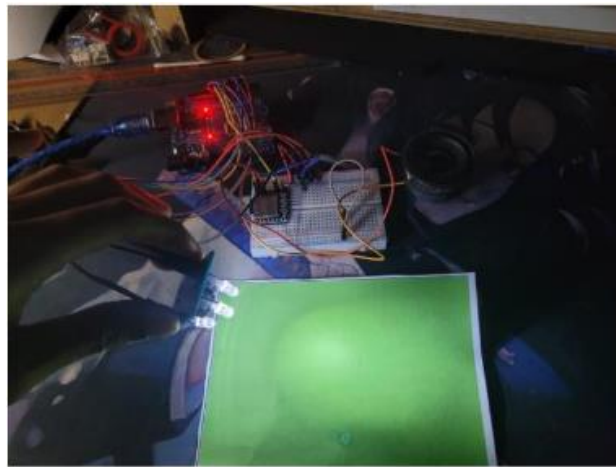


Figure 7. Green Color Detection Process.
(Source: Personal Documentation, 2025)

Next, Figure 7 shows the green color detection process, with the sensor reading the characteristic R and B values according to the thresholds (R: 40–55 and B: 5–25). The system provides an audio notification via the DFPlayer Mini with the file 001.mp3 to indicate that the green color has been detected.

4. Conclusion

This research successfully developed an automatic color detection system using the TCS3200 sensor connected to an Arduino Uno, demonstrating satisfactory results. The system can recognize three primary colors: red, green, and blue, with high accuracy through a frequency-based thresholding algorithm. The calibrated threshold values are stable, with specific detection ranges for each color. Integration with the DFPlayer Mini provides fast and consistent audio feedback. Validation on the Wokwi platform demonstrated stable system performance before real hardware implementation, helping to reduce prototype development time and cost. This system has potential applications in various fields, such as industrial automation and assistive technology for visually impaired individuals. Recommendations for further development include expanding the number of detectable colors, implementing an automatic calibration system, improving the user interface, optimizing power consumption, studying environmental factors, developing a color database, incorporating wireless connectivity, and conducting field testing to validate system performance.

References

- Chaudry, A. M. (2020). Using Arduino Uno Microcontroller to Create Interest in Physics. *The Physics Teacher*, 58(6), 418-421. <https://doi.org/10.1119/10.0001841>
- Chowdhury, I., & Ahmed, T. (2021). Design and Prototyping of Sensor-based Anti-Theft Security System using Microcontroller. 10(03), 58-66. <https://doi.org/10.17577/IJERTV10IS030019>
- Duhi, S. (2025). Sistem Monitoring Lokasi dan Data Gangguan Satu Fase Menggunakan Esp8266 via Blynk dan Telegram Location and Single-Phase Fault Data Monitoring System Using ESP8266 via Blynk and Telegram. 7.

- Hadiati, S., Bely, K., Pramuda, A., & Angraeni, L. (2023). Bibliometric Analysis of Microcontroller-Based Learning Media. 9(2), 152-160. <https://doi.org/10.25273/jpfk.v9i2.20945>
- Jie, L. W., Sen, T. P., Maniha, N., Ghani, A., & Abas, M. F. (2021). Automatic Control of Color Sorting and Pick / Place of a 6- DOF Robot Arm. 54(3), 435-443. <https://doi.org/10.18280/jesa.540306>
- Khanh, H. Q., Ngon, N. C., Nguyen, T., Lan, P., Nguyen, V., & Phuc, H. (2022). Evaluating the optimal working parameters of the color sensor TCS3200 in the fresh chili destemming system. February 2023. <https://doi.org/10.22144/ctu.jen.2022.004>
- Nuratch, S. (2017). A universal microcontroller circuit and firmware design and implementation for IoT-based real-time measurement and control applications. 2017 International Electrical Engineering Congress, IEECON 2017. <https://doi.org/10.1109/IEECON.2017.8075906>
- Pembangunan, U., Budi, P., & Sumatera, N. (2025). Automatic Classification System Color Of Bottle Detection Tool Using TCS3200 Sensor Based On Arduino. 13(02), 452-464. <https://doi.org/10.58471/infokum.v13i02>
- Penderia, P., Tcs, W., & Uno, A. (2024). Development of TCS3200 Colour Sensor based on Arduino Uno Microcontroller for Determination of Capsaicin Level in Sauces. 53(12), 3339-3348. <https://doi.org/10.17576/jsm-2024-5312-17>
- Raikar, M. M. (2023). Color based Classification of Products Using Internet of Things. November. <https://doi.org/10.1007/978-981-99-2710-4>
- Revansa Arya Pradhana, D., Ichsan Pradhana, A., & Hartanti, D. (2024). Implementasi Internet Of Things Pada Sistem Kontrol RC-car Pemantau Area Berbasis ESP32 cam. Jurnal Teknologi, 11(2), 149-159. <https://doi.org/10.31479/jtek.v11i2.306>
- Swathi, K., Sandeep, T. U., & Ramani, A. R. (2018). Performance Analysis of Microcontrollers Used In IoT Technology. International Journal of Scientific Research in Science, Engineering and Technology, 4(4), 1268-1273.
- Timer Modes for ATmega328P. (n.d.). 0-2.
- Vicananda, L. P., Prahasti, S. N., Maruf, K., & Setiawan, R. J. (2024). Design and Implementation of a Candy Color Sorter Device using Microcontroller-Based Color Sensor TCS3200. 8(1), 1-8. <https://doi.org/10.18196/jet.v8i1.20346>
- Wadly, F., Ramadhan, Z., & Rosan, K. (2025). Integrating the NodeMCU ESP8266 Microcontroller with the DHT11 and MQ-135 Sensors Enables IoT-Based Air Quality Monitoring. International Conference on Artificial Intelligence, Navigation, Engineering, and Aviation Technology, 2(1), 272-277.