STM JOURNALS

International Journal of
Advanced Robotics and
Automation Technology

Volume 2, Issue 2, 2024
July—December
DOI (Journal): 10.37591/IJARAT

https://journals.stmjournals.com/ijarat

Research IJARAT

Arduino based Colour Sensing System with Real-time RGBDisplay on LCD

Gourav Vivek Kulkarni[1,*], P. Brisli Jacob[2], Akshay P. Rao[3], Raksha Sai Rannorey[4], Nidhi B.C.[5], Mayura S Yogi[6]

Abstract

Using a microcontroller Arduino UNO and TCS3200 color sensors, this project investigates the creation of a color measuring system. Colour sensors operate by emitting light onto an object and analyzing the reflected wavelengths to determine its colour. Our system is designed to detect a wide range of colours with high precision and display the corresponding Red-Green-Blue (RGB) values in real-time on a liquid crystal display (LCD). The TCS3200 sensor, developed by Texas Advanced Optoelectronic Solutions Inc., is central to this system. It includes an array of photodiodes and uses white LEDs to illuminate the target object. The sensor may measure the amount of red, green, and blue light reflected from an object by choosing the right photodiode array. The Arduino UNO processes it and then shows the RGB values it noticed on the LCD. In addition to the primary functionality of colour detection, the system is designed for future integration with a WIFI module to enable wireless communication and expand its applications. This project demonstrates the practical application of Arduino-based systems in tasks such as object sorting, quality control, and automation, highlighting the versatility and effectiveness of using affordable colour sensors in various industrial and consumer applications.

Keywords: Colour Sensor, Arduino UNO, TCS3200, RGB Detection, Liquid Crystal Display (LCD)

INTRODUCTION

Introduction

A Colour Sensor is a device designed to identify and analyse colours by emitting light onto an object and then detecting and interpreting the reflected light. Typically, these sensors utilize external light sources such as arrays of white LEDs to illuminate the objectunder observation. By analysing the wavelengths and intensity of the reflected light, colour sensors can accurately determine the colour of the object. The applications of colour sensors are diverse, ranging from sorting objects based on colour to quality control systems and enhancing colour accuracy in printers.

In our project, we have implemented a straightforward Arduino Colour Sensor application utilizing TCS3200 colour sensors. These sensors have the knack of accurately detecting a large spectrum of wavelengths.

By integrating these sensors with Arduino, we have developed a versatile system capable of accurately identifying different colours. The TCS3200 colour sensors provide reliable data, enabling the application to perform tasks such as

*__Author for Correspondence__
Gourav Vivek Kulkarni
E-mail: gouravvivekk.mmd22@rvce.edu.in

[1-6]Student, Department of Mechanical Engineering, RV College of Engineering, Bengaluru, Karnataka, India

Received Date: October 29, 2024
Accepted Date: November 11, 2024
Published Date: November 20, 2024

Citation: Gourav Vivek Kulkarni, P. Brisli Jacob, Akshay P. Rao, Raksha Sai Rannorey, Nidhi B.C., Mayura S Yogi. Energy-Efficient HVAC System with Decision Tree Classifier and Real-Time SMS Notification. International Journal of Advanced Robotics and Automation Technology. 2024; 2(2): 35–43p.

sorting objects by colour or enhancing colour quality in various applications. This project showcases the effectiveness and simplicity of incorporating colour sensors into Arduino-based systems for colour detection purposes [1].

Methodology

1. Component Selection and Setup:
   - *Arduino Microcontroller:* Choose an Arduino board (such as Arduino Uno) to act as the main control unit for the system.
   - *Color Sensor (TCS3200 or TCS34725):* Select a color sensor capable of detecting RGB values. These sensors contain photodiodes with different filters that allow them to sense red, green, and blue light levels in the environment.
   - *LCD Display:* Use a 16x2 or 20x4 LCD display module to show the RGB values in real-time. Connect it to the Arduino for output display.
   - *Other Components:* Use jumper wires, a breadboard, and resistors as needed for connections (Figure 1).

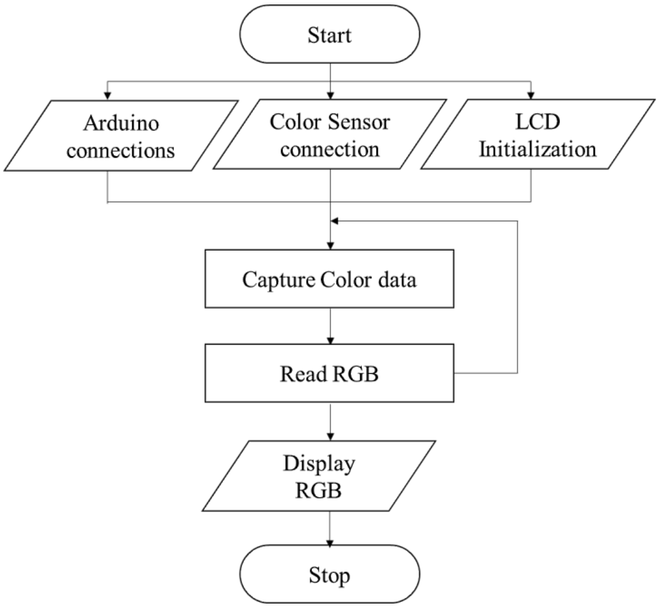

Figure 1. Methodology.

2. Circuit Design:
   - Connect the **color sensor** to the Arduino according to the sensor's specifications (typically involving connections for power, ground, and digital output pins).
   - Connect the **LCD display** to the Arduino using appropriate pins (e.g., for I2C, connect to SDA and SCL pins, or use designated digital pins for other types of LCDs).
   - Ensure that all connections are secure, with the color sensor positioned to receive accurate light input from the target object [2,3].
3. Software Development
   - Use the **Arduino IDE** for coding.
   - Install any necessary **libraries** for the color sensor and LCD display (such as the TCS3200 or TCS34725 library for color sensors and the LiquidCrystal library for LCD displays).
   - Write code to:
     - **Initialize** the sensor and LCD display.
     - **Read RGB values** from the color sensor in real-time.
     - **Convert sensor readings** into meaningful RGB values that can be displayed.
     - Display the RGB values on the LCD screen.

- Implement a loop that continuously reads data from the sensor and updates the display in real-time.

4. Calibration:
    - Place a white object in front of the color sensor and read the raw RGB values. Use these values to set a baseline for color detection.
    - If necessary, adjust the code to **normalize the RGB values** based on the baseline readings to improve color accuracy.
5. Testing and Debugging:
    - Test the system by placing different colored objects in front of the color sensor.
    - Observe the RGB values displayed on the LCD to ensure they are consistent with the colors being detected.
    - Make adjustments to the code if the sensor readings do not match expected values, adjusting for ambient light conditions as needed.
6. System Optimization:
    - Optimize the code for smoother performance and reduce any delay in updating RGB values on the LCD display.
    - If necessary, add a filter in the code to reduce fluctuations in sensor readings due to changes in ambient light or shadows.
7. Final Testing and Validation:
    - Test the system in various lighting conditions to ensure that it accurately displays RGB values across different environments.
    - Validate the color readings against known RGB values to verify the system's accuracy and reliability.

Objective

The objective of the project is to integrate an Arduino UNO microcontroller with a colour sensorand a liquid crystal display (LCD) to accurately detect the colour of a specified object. The primary objective is to develop a system capable of sensing the colour of an object and displayingits constituent Red-Green-Blue (RGB) values in real-time on the LCD display. To further enhance the project's capabilities, integration with a WIFI module will be pursued in future scope work. This addition will enable the project to achieve a logical end by facilitating wireless communication and potentially expanding its functionality and applicability [4].

Color Sensor

A colour sensor is a straightforward tool capable of identifying the colour of an object, enabling subsequent actions based on the detection. It offers convenient solutions for sorting and packaging tasks in industrial settings. While pricier sensors are common in industries, more affordable options like the TCS3200 serve well for less complex applications [5].

Parts of Colour Sensor

- This sensor identifies the colour and puts the result out in the digital format. The parts marking of the TCS3200 colour sensor is shown in Figure 2
- The key element of the sensor is the TS3200 Color Sensor IC, which was created and developed by Texas Advanced Optoelectronic Solutions Inc. and is highly affordable and user-friendly.
- There are also four white LEDs, these LEDs light up when the module powers on. TheseLEDs light up the object that the sensor needs to sense.
- There is a filter capacitor, a decoupling capacitor, and some resistors onboard. This module operates between 2.7 and 5.5 volts.

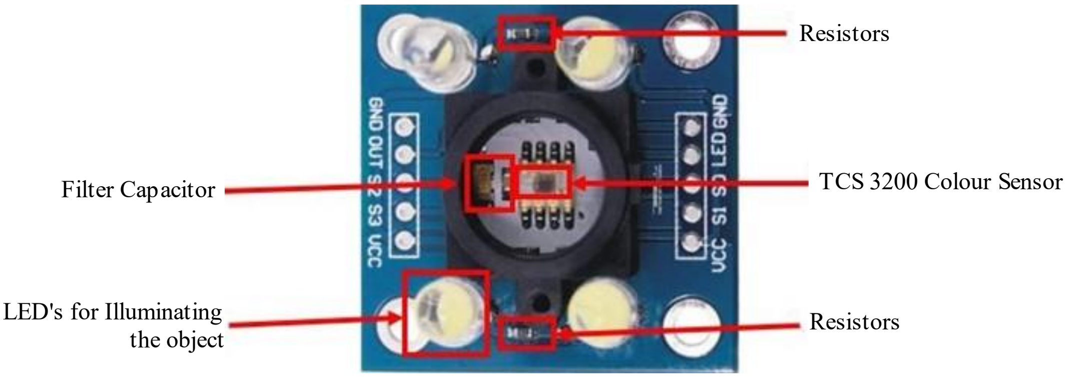

Figure 2. Parts of Colour Sensor.

Pin Designation of Colour Sensor

- *VCC:* VCC is the hue sensor's power supply pin, which can be wired to both the supply's 3.3V or 5V.
- *S0 and S1:* The sensor's Return Frequency Scaling Percentage can be altered chosen using the S0 and S1 pins. It may be changed to 2%, 20%, or 100% scaling by manipulating these pins.
- *S2 and S3:* The S2 & S3 pins can be used to select the colour array of the sensor. Byselecting the right colour array one after the other, this sensor identifies a colour.
- *OUT:* This is the sensor's output pin. As the sensor detects a specific color, the output pulse velocity on this pin changes. We can identify the color by observing this change in wave width [6].
- *GND:* Ground is the Rgb Sensor module's ground pin, and it need to be linked to the Arduino's main pin (Figure 3).

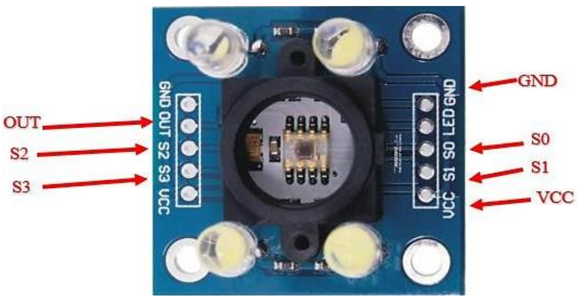

Figure 3. Pin Designation.

Working of Colour Sensor

The colour sensor operates on a fundamental principle of light. Red, green, and blue are the three of them main colors that make up white light, and each has a particular wavelength.

When light interacts with a surface, it either gets absorbed or reflected based on the material's properties. Our eyes perceivethe reflected light, allowing us to see different colours.

In order to set our color sensor to 20% frequency scaling, we are sending high and low signals to the S0 and S1 pins of the camera module. This indicates that a frequency of 200 KHz will be used by the sensor. In this mode, the sensor's output frequency will vary based on the color value, but its duty cycle will stay at 50%. The array of photodiodes is selected by combining the S2 and S3 pins. As you can see in the animation, we will change the array to red, green, and blue filter one by one for each color that the sensor is able to identify (Table 1 & Table 2).

The output frequency of the sensing will rise if the color it detects matches the array it has chosen. In all other cases, the output frequency either remains comparatively steady or shows a slight fluctuation [7].

Table 1. S0 and S1 Sets Output Frequency Scaling Factor:

S0	S1	Output Frequency Scaling Factor
L	L	Power down mode
L	H	Scaling Factor 2%
H	L	Scaling Factor 20%
H	H	Scaling Factor 100%

Table 2. S2 and S3 Sets the Internal Filter Type:

S2	S3	Output Frequency Scaling Factor
L	L	Red Filter
L	H	Blue Filter
H	L	Clear (No Filter)
H	H	Green Filter

CIRCUIT DIAGRAM

Circuit

Components Needed
Arduino Uno (or similar)
Color Sensor (e.g., TCS3200 or TCS34725)
LCD Display (16x2 or 20x4 with I2C module if available, or a standard parallel LCD)
Breadboard (optional)
Jumper Wires (Figure 4)

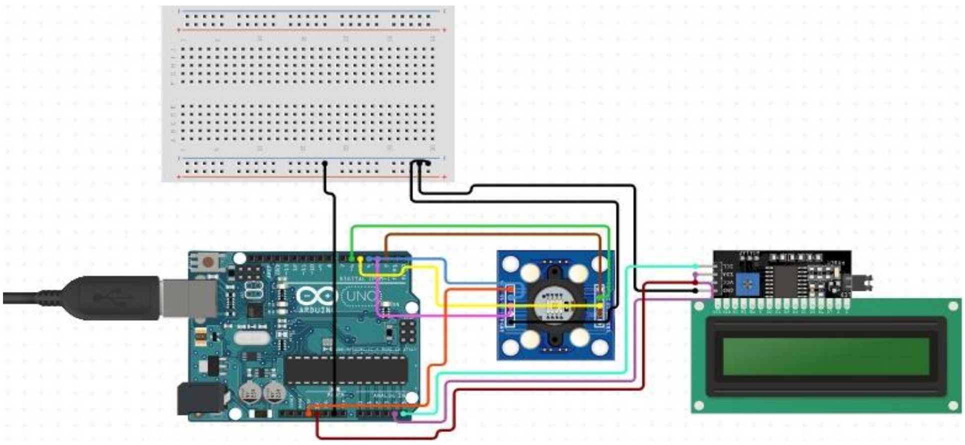

Figure 4. connection diagram.

CODING THE MICROPROCESSOR USING ARDUINO IDE

Code Used

```
#include "stationDefines.h"#include <Wire.h>
#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()

{
```



```
pinMode(S0, OUTPUT); pinMode(S1, OUTPUT); pinMode(S2, OUTPUT); pinMode(S3,
OUTPUT);
pinMode(outPin, INPUT); //out from sensor becomes input to arduino
// Setting frequency scaling to 100%digitalWrite(S0,HIGH); digitalWrite(S1,HIGH);

Serial.begin(9600);lcd.init(); lcd.backlight();
lcd.print("IIOT-Lab EL");Serial.println("IOT"); startTiming = millis();
}
void loop()
{
getColor();
if (DEBUG) printData(); elapsedTime = millis()-startTiming;if (elapsedTime > 1000)
{
showDataLCD();
startTiming = millis();
}
}
void getColor()
{
readRGB();
if (red > 210 && red < 240 && grn > 140 && grn < 175 && blu > 150 && blu < 185)colour =
"SCARLET";
else if (red > 220 && red < 237 && grn > 150 && grn < 175 && blu > 175 && blu < 200)colour
= "CANDY";
else if (red > 155 && red < 185 && grn > 135 && grn < 165 && blu > 140 && blu < 165)colour
= "FERERO";
else if (red > 205 && red < 225 && grn > 170 && grn < 200 && blu > 190 && blu < 225)colour
= "LAVENGER";
else if (red > 240 && red < 290 && grn > 240 && grn < 290 && blu > 240 && blu < 290)colour
= "WHITE";
else if (red > 0 && red < 140 && grn > 0 && grn < 140 && blu > 0 && blu < 140) colour =
"BLACK";
else colour = "NO_COLOR";
}
/* read RGB components */void readRGB()
{
red = 0;
grn = 0;
blu = 0; int n = 10;
for (int i = 0; i < n; ++i)
{
//read red component digitalWrite(s2, LOW);digitalWrite(s3, LOW);
red = red+pulseIn(outPin, LOW);
//read green component digitalWrite(s2, HIGH);digitalWrite(s3, HIGH);
grn = grn+pulseIn(outPin, LOW);

//read blue component digitalWrite(s2, LOW);digitalWrite(s3, HIGH);
blu = blu+pulseIn(outPin, LOW);
}
red=305-red;grn=305-grn;blu=305-blu;
}
```



```
void showDataLCD(void)
{
lcd.clear(); lcd.setCursor (0,0); lcd.print("R"); lcd.setCursor (1,0); lcd.print("    "); lcd.setCursor (1,0); lcd.print(red); lcd.setCursor (5,0); lcd.print(" G"); lcd.setCursor (7,0); lcd.print("     "); lcd.setCursor (7,0); lcd.print(grn); lcd.setCursor (12,0);lcd.print("B"); lcd.setCursor (13,0);lcd.print(" "); lcd.setCursor (13,0);
lcd.print(blu); lcd.setCursor (0,1); lcd.print("Colour: ");lcd.setCursor (7,1); lcd.print("   "); lcd.setCursor (7,1); lcd.print(colour);
}
void printData(void)
{
Serial.print("red= "); Serial.print(red); Serial.print(" green= "); Serial.print(grn); Serial.print(" blue= "); Serial.print(blu); Serial.print (" - "); Serial.print (colour); Serial.println (" detected!");
}
#define S0 2
#define S1 4
#define S2 7
#define S3 13
#define outPin 8 boolean DEBUG = true;
// Variablesint red;
int grn;int blu;
String colour ="";int count = 0;
long startTiming = 0;long elapsedTime =0;
```

RESULTS AND CONCLUSION

The results of the "Arduino-based Colour Sensing System with Real-time RGB Display on LCD" project demonstrate the system's ability to accurately detect and display the RGB color values of various objects in real time (Figures 5 and 6).

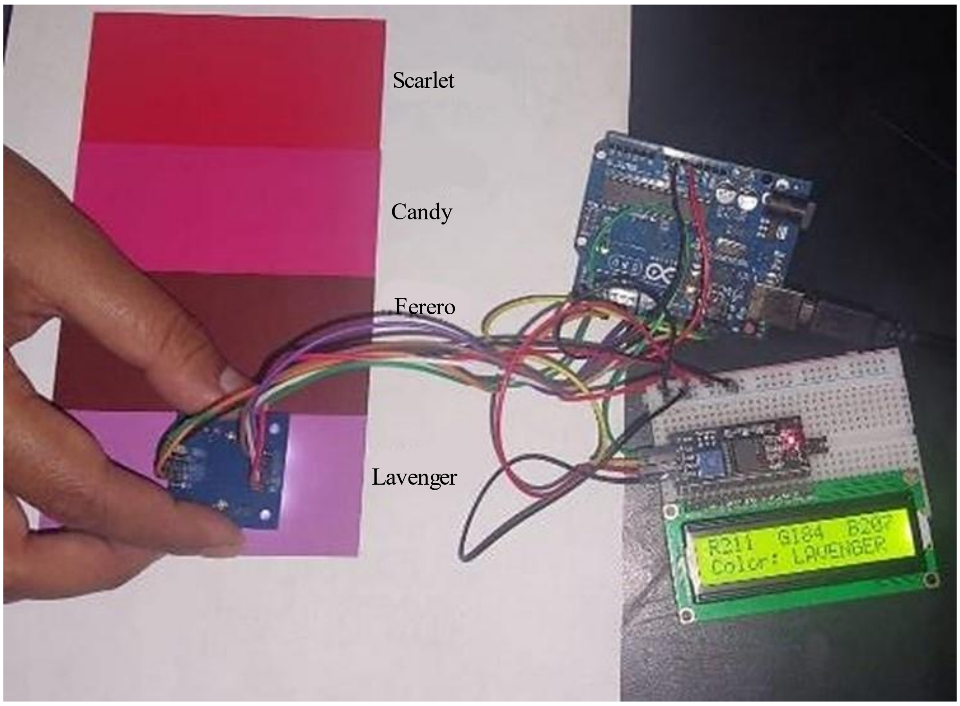

Figure 5. Result example 1.

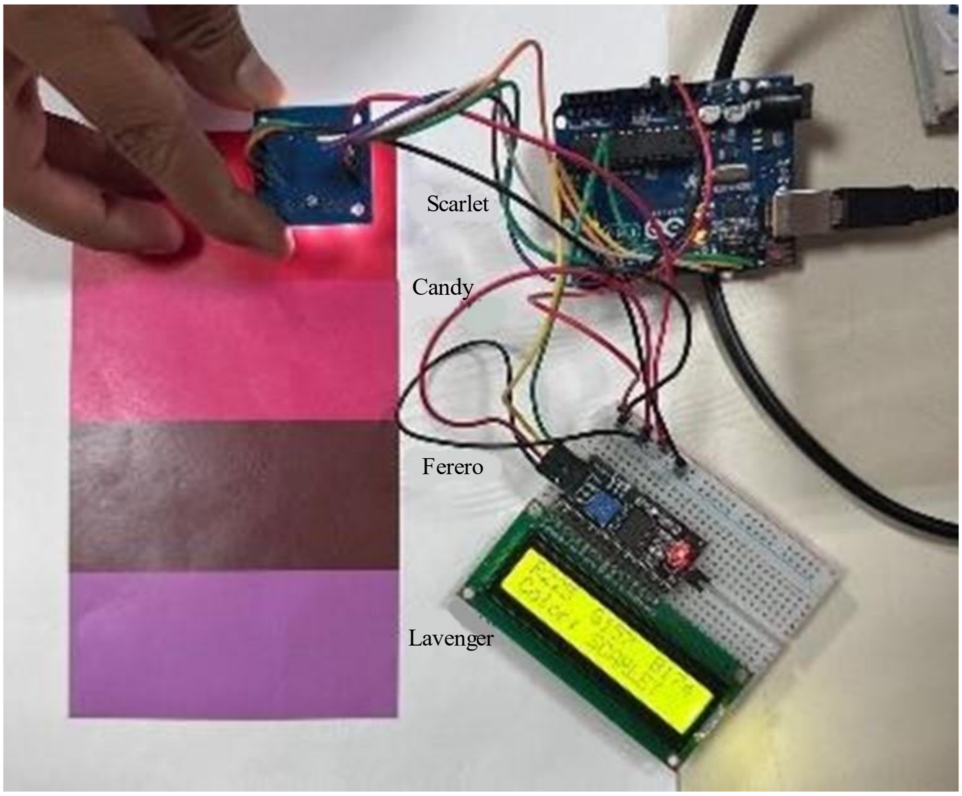

Figure 6. Result example 2.

CONCLUSION

In conclusion, the integration of Arduino UNO with a colour sensor and liquid crystal display (LCD) has proven to be an effective method for sensing the colour of a particular object and displaying its constituent RGB values. Through the implementation of the provided Arduino code, we were able to accurately detect colours and display the corresponding RGB values on the LCD display. This EL project has demonstrated the practical application of Arduino in colour detection and data visualization, showcasing its versatility in interfacing with external components such as colour sensors and displays. By leveraging this technology, users can easily develop custom solutions for colour sensing and analysis in various applications, including automation, robotics, and smart devices.

Future Scope

- *Colour Shade Quality Assessment:* Implement a feature to assess the quality of the detected colour shade by comparing it to an actual formulated shade. This will ensure more accurate colour matching and enhance the reliability of the system in cosmetic applications.
- *Integration with Cloud-Based colour Sorting Equipment:* Explore integration with cloud-based colour sorting equipment to enable remote monitoring and management of colour detection processes. This integration will facilitate real-time data analysis and decision-making, improving efficiency and scalability.
- *Expansion into Customized Cosmetic Solutions:* Extend the project to develop customized shade-providing equipment for various cosmetics, including lipstick, foundation, and makeup kits. By integrating the WIFI module, users can remotely select and customize their desired shades, offering a personalized experience

REFERENCES

1. Circuitdigest.com. (n.d.). How Does the TCS3200 Color Sensor Work and how toInterface it with Arduino? [online] Available at: https://circuitdigest.com/microcontroller-projects/interfacing-color-sensor-with- arduino.

2. TCS3200, TCS3210 programmable color light-to-frequency converter S0 1 S1 2 OE 3 GND 4 TCS3210. (2009). Available at: https://components101.com/sites/default/files/component_datasheet/TCS3200-Color- Sensor-Module.pdf [Accessed 28 Mar. 2024].
3. Hackster.io. (n.d.). TCS3200 Color Sensor Arduino and LCD display circuit. [online] Available at: https://www.hackster.io/jack100/tcs3200-color-sensor-arduino-and-lcd- display-circuit-72a9d5#:~:text=The%20TCS3200%20color%20sensor%20is [Accessed 28 Mar. 2024].
4. Moch. Azriel Revanza, Syamsudduha Syahrorini and Akhmad Ahfas (2023). Freshness Detection And Sorting Of Pears Using The TCS-3200 Sensor. Journal of Computer Networks, Architecture and High Performance Computing, 5(2), pp.482–493. doi:https://doi.org/10.47709/cnahpc.v5i2.2470.
5. Dhawan, A., Kaushal, P. and Rohini Mudhalwadkar (2021). Development of Low Cost and Portable Colorimeter Using TCS 3200 Color Sensor. 2021 Asian Conference on Innovation in Technology (ASIANCON). doi:https://doi.org/10.1109/asiancon51346.2021.9544866.
6. Mohamad Habib Ahsan, Arief Marwanto and Agus Suprajitno (2018). Colour Detector Tool Using Tcs3200 And Arduino Uno For Blind And Child. 6(1), pp.37–44.
7. Badamasi, Y.A. (2014). The working principle of an Arduino. 2014 11th International Conference on Electronics, Computer and Computation (ICECCO). [online] doi:https://doi.org/10.1109/icecco.2014.6997578.