

COLOUR BASED PRODUCT SORTING

A MINI PROJECT REPORT

Project done by

JACK CHIN S
B.E. ECE

| TABLE OF CONTENTS | | | |
|-------------------|-----------------------------------|-------------------------|----------|
| Sl.No | TITLE | | PAGE NO. |
| 1. | ABSTRACT | | 1 |
| 2. | INTRODUCTION | | 2 |
| | 2.1 | PROBLEM STATEMENT | |
| | 2.2 | OBJECTIVE | |
| | 2.3 | CHALLENGES | |
| 3. | LITERATURE REVIEW/EXISTING SYSTEM | | 5 |
| 4. | SYSTEM OVERVIEW/METHODOLOGY | | 7 |
| 5. | 5.1 | HARDWARE REQUIREMENTS | 9 |
| | 5.2 | SOFTWARE REQUIREMENTS | |
| 6. | CIRCUIT DIAGRAM & CONNECTIONS | | 14 |
| 7. | PROGRAM | | 17 |
| 8. | 8.1 | IMPLEMENTATION | 21 |
| | 8.2 | WORKING | |
| | 8.3 | PROTOTYPE/WORKING MODEL | |
| 9. | 9.1 | APPLICATIONS | 23 |
| | 9.2 | LIMITATIONS | |
| | 9.3 | FUTURE SCOPE | |
| 10. | 10.1 | CONCLUSION | 25 |
| | 10.2 | REFERENCES | |

ABSTRACT

“COLOUR BASED PRODUCT SORTING”

In the modern era of industrial automation and smart manufacturing, color-based product sorting plays a significant role in quality control, packaging, and classification of goods. This project presents the design and implementation of a Color-Based Product Sorting System using an Arduino Pro Mini (ATmega328P) microcontroller, a TCS3200 color sensor, a 0.96" I2C OLED display, and two servo motors. The objective is to develop a compact, low-cost, and efficient system that can detect the color of an object and accordingly sort it into designated categories.

The core sensing component of the system is the TCS3200 color sensor, which can identify colors based on the intensity of reflected light. It outputs frequency signals proportional to the red, green, and blue components of the object being scanned. The microcontroller processes this data to classify the color into predefined categories such as Red, Green, Yellow, White, and Black.

To enhance the user interface and provide real-time feedback, a 0.96" OLED display (I2C, 4-pin) is used to show the identified color. This allows users to monitor the system's operation and ensures transparency in sorting.

The system uses two servo motors to control mechanical arms or gates. One servo is used to position the scanning platform or sensor over the product, and the other servo redirects the product to its respective bin based on the detected color. This dual-servo arrangement ensures accurate and dynamic sorting with minimal mechanical complexity.

The entire system is compact and powered by the ATmega328P microcontroller, which efficiently manages sensor data acquisition, decision-making, and actuator control. It also ensures low power consumption, making the solution suitable for embedded and portable systems.

This project serves as a practical demonstration of how microcontrollers, sensors, and actuators can be integrated to automate real-world industrial processes. It highlights the importance of embedded systems in smart manufacturing and provides a foundation for future development in more complex vision-based sorting technologies.

CHAPTER - 2

INTRODUCTION

In today's fast-paced industrial and packaging sectors, efficient product sorting has become a vital component in achieving high productivity, accuracy, and cost-effectiveness. Conventional sorting systems often involve manual processes or high-end machines, which can be labor-intensive, error-prone, and financially burdensome for small to medium-sized enterprises. With the advancement of embedded systems and sensor technologies, compact and cost-effective automated sorting solutions have emerged, enabling a new generation of intelligent sorting mechanisms.

This project explores the development of a color-based product sorting system using the Arduino Pro Mini (ATmega328P) microcontroller, TCS3200 color sensor, 0.96" I2C OLED display, and servo motors. The system is designed to detect the color of an object and sort it into a specific category using mechanical actuators controlled by the microcontroller. The Pro Mini, known for its low power consumption and small footprint, serves as the core processing unit responsible for sensor data acquisition, decision making, and servo control.

Color recognition is achieved using the TCS3200 color sensor, which identifies objects based on the intensity of reflected light. This sensor converts light information into frequency signals corresponding to red, green, and blue components. The microcontroller processes these values to determine the color of the object. Identified colors such as Red, Green, Yellow, White, and Black are displayed in real-time on a 0.96-inch OLED display via the I2C communication protocol, offering instant user feedback and visual confirmation.

Two servo motors are integrated into the system—one to guide the object in position for color detection, and the other to control the sorting mechanism that directs objects to the appropriate bins. This motorized setup ensures accurate and efficient sorting with minimal mechanical complexity.

Unlike image processing systems that require high computational resources, this embedded solution offers simplicity, affordability, and reliability. It is well-suited for educational purposes, small-scale automation, and prototyping environments. The combination of the Pro Mini microcontroller, color sensing, real-time display, and mechanical control provides a strong foundation for basic industrial automation projects.

This report details the design principles, working methodology, and implementation of the color-based sorting system. It also discusses the benefits, applications, and potential enhancements of using embedded platforms for intelligent product classification. The project highlights how low-cost, easily programmable hardware can be effectively used to automate repetitive tasks and improve operational efficiency in various domains.

2.1 PROBLEM STATEMENT:

Manual sorting of products based on color in fields like agriculture, packaging, and retail is often slow, inconsistent, and labor-intensive. This project aims to address these issues by developing an automated color-based sorting system using a Pro Mini ATmega328P microcontroller, TCS3200 color sensor, a 0.96" I2C OLED display, and two servo motors. The system identifies object colors and sorts them accordingly, significantly reducing human effort while increasing speed and accuracy. It offers a low-cost, reliable solution for small-scale automation where precision and efficiency are essential.

2.2 OBJECTIVE:

The main objective of this project is to design and implement an automated system capable of detecting and sorting products based on their color using a cost-effective and compact microcontroller-based setup. The system utilizes a TCS3200 color sensor to identify the color of objects, a Pro Mini ATmega328P to process the sensor data, an OLED I2C display to show the detected color in real-time, and two servo motors to physically sort the objects into designated categories. This project aims to minimize human intervention, reduce sorting errors, and enhance efficiency in color-based segregation tasks for applications in agriculture, packaging, and small-scale manufacturing.

2.3 CHALLENGES:

- **Inconsistent Ambient Lighting:**
Changes in surrounding light can affect the color sensor's readings, leading to inaccurate color detection.
- **Limited Microcontroller Resources:**
The Pro Mini ATmega328P has limited memory and processing capability, which restricts handling multiple components and complex logic simultaneously.
- **Servo Motor Accuracy:**
Achieving precise and repeatable servo movements for correct sorting requires careful calibration and may be affected by load or voltage variations.
- **Color Threshold Tuning:**
Differentiating between closely related colors (e.g., red and orange) requires accurate threshold values, which need careful testing and adjustment.
- **Mechanical Alignment of Objects:**
Improper positioning of objects in front of the sensor may lead to incorrect color detection and sorting errors.

CHAPTER 3:

LITERATURE REVIEW / EXISTING SYSTEM

In the current landscape of automation, several color-based sorting systems have been developed and implemented using various technologies. Traditionally, color sorting machines are widely used in industries such as agriculture (for sorting grains, fruits, and vegetables), manufacturing (to sort colored components), and recycling plants (to segregate waste based on color). These systems often rely on expensive and complex vision-based technologies using high-resolution cameras and image processing algorithms powered by platforms such as Raspberry Pi or full-scale computers.

Existing systems commonly use **Raspberry Pi** or **Arduino Uno** along with camera modules and OpenCV libraries for real-time image processing. While these systems offer high accuracy, they also involve higher power consumption, cost, and complexity in programming and hardware integration.

In contrast, the proposed system uses a more **cost-effective and simplified approach** by utilizing the **TCS3200 color sensor**, which is capable of detecting basic colors through direct light frequency analysis. The system is built around the **Pro Mini ATmega328P microcontroller**, known for its compact size and low power consumption, making it ideal for small-scale or embedded sorting applications.

Furthermore, the addition of a **0.96" I2C OLED display** allows for real-time display of the identified color, improving the transparency and debugging process. The sorting mechanism is executed through **two servo motors**, which physically direct the object to a specific location based on the detected color.

Compared to existing high-end systems, this approach is **more affordable, compact, and accessible**, particularly suitable for educational, laboratory-scale, or low-volume industrial applications. It simplifies color-based product sorting without requiring complex image processing, making automation more approachable for small businesses and learners.

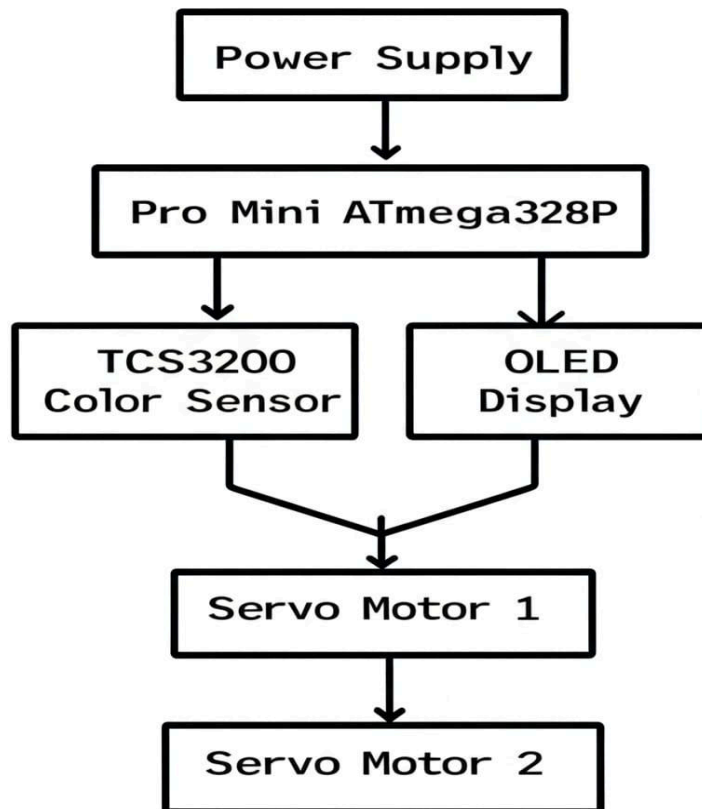
Limitations of Existing Systems:

- **High Cost:** Systems using Raspberry Pi and cameras with image processing are expensive and not feasible for small-scale applications.
- **Complex Setup:** Image processing-based systems require extensive coding, calibration, and tuning, which increases development time and complexity.
- **High Power Consumption:** Advanced processors and camera modules consume more power, making them unsuitable for battery-operated or portable setups.
- **Bulky Hardware:** Existing systems often require large components, reducing portability and integration into compact environments.
- **Slower Processing for Basic Tasks:** Using high-end processing for simple color detection is often overkill and inefficient.

CHAPTER 4:

SYSTEM OVERVIEW/METHODOLOGY

Block Diagram/Flowchart of the system:



Description of Working Principle:

The system uses the **TCS3200 color sensor** to detect the color of a product placed in front of it. Once the color is detected, it is processed by the **ATmega328P Pro Mini microcontroller**. Based on the identified color, a corresponding message is displayed on the **0.96" I2C OLED display**, and a pair of **servo motors** actuate to move the product into the correct bin or direction for sorting.

The TCS3200 outputs frequency signals based on color intensity. The microcontroller counts the pulse durations for red, green, and blue filters, compares them, and determines the dominant color. Servo motors then rotate to predefined angles corresponding to specific color bins.

Step-by-Step Methodology:

➤ System Initialization:

The Pro Mini initializes the TCS3200 color sensor, OLED display, and configures the servo motors.

➤ Color Detection:

The object is placed in front of the color sensor. The sensor emits light, reads reflected light intensities, and sends frequency signals for RGB components to the microcontroller.

➤ Color Analysis:

The ATmega328P processes the RGB frequency values and determines the object's color (e.g., Red, Green, Yellow, White, or Black).

➤ Display Output:

The identified color is displayed in real-time on the OLED screen via the I2C interface.

➤ Servo Motor Activation:

Based on the detected color:

- **Servo 1** positions the sorting ramp.

- **Servo 2** may act as a pusher or gate to drop the object into the correct bin.

➤ Repeat Cycle:

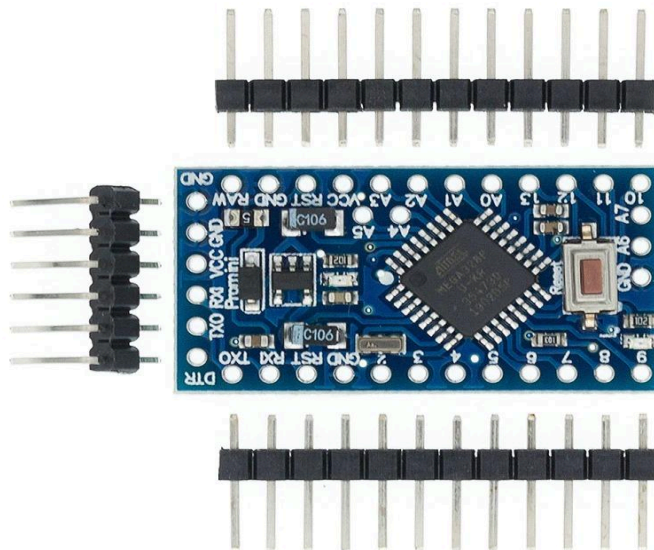
The system resets and waits for the next object.

CHAPTER - 5

5.1 HARDWARE REQUIREMENTS:

- **Arduino Pro Mini (ATmega328P)**

- Acts as the central microcontroller unit (MCU).
- Controls sensor readings, logic decisions, display output, and servo movements.
- Compact and power-efficient board ideal for embedded applications.



- **TCS3200 Color Sensor**

- Used to detect and identify the color of objects.
- Converts light intensity to frequency using photodiodes and a current-to-frequency converter.



- **0.96" OLED Display (I2C, 4-Pin)**

- Displays the identified color name in real-time.
- Communicates via the I2C protocol with just 2 data pins (SDA, SCL).
- Helps in debugging and user interaction.



- **Servo Motors (x2)**

- Used for mechanical actuation in sorting.
 - **Servo 1:** Directs the ramp or gate.
 - **Servo 2:** Pushes or releases the item into the designated bin.
- Controlled via PWM signals from the Pro Mini.



- **Power Supply (5V regulated)**

- Provides necessary voltage and current to drive the Pro Mini, sensor, display, and servos.
- Could be a battery pack or DC adapter with a voltage regulator.

- **Connecting Wires and Breadboard / PCB**

- For prototyping and circuit connections.
- Ensures reliable and organized wiring between components.

- **Mounting Frame / Stand**

- Holds the sensor and object in place during detection.
- Guides objects along the path for sorting.

- **Optional: FTDI USB to Serial Adapter**

- Required to upload code to the Pro Mini, which lacks an onboard USB port.

5.2 SOFTWARE REQUIREMENTS:

➤ Arduino IDE

- Main platform used to write, compile, and upload the code to the Pro Mini.
- Supports C/C++ based programming tailored for microcontrollers like ATmega328P.
- Compatible with libraries for OLED, Servo, and TCS3200.

➤ TCS3200 Color Sensor Library

- A custom or open-source library to simplify interfacing with the TCS3200 color sensor.
- Handles color frequency readings and conversion to RGB or named colors.
- Can be replaced with manual signal reading logic if needed.

➤ Adafruit SSD1306 & GFX Library

- Required for displaying text or data on the 0.96" OLED I2C display.
- *Adafruit_GFX.h* provides graphics primitives.
- *Adafruit_SSD1306.h* supports the OLED controller (SSD1306).
- Installed through Arduino Library Manager.

➤ Servo Library

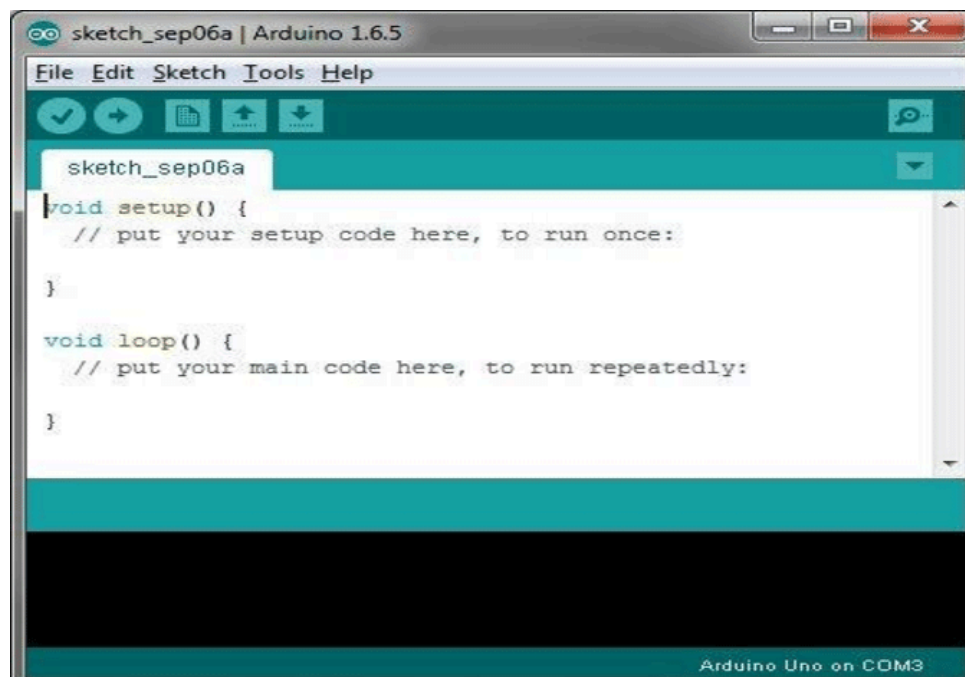
- *Servo.h* helps in generating PWM signals to control servo motors.
- Provides simple *write(angle)* functions to move servos accurately.

➤ FTDI USB Driver (for Pro Mini Programming)

- Needed to establish communication between the computer and Pro Mini via FTDI USB to Serial adapter.
- Drivers are usually auto-installed, but may need manual setup depending on the OS.

➤ Serial Monitor (Optional)

- Built into Arduino IDE.
- Used for debugging and testing sensor output or logic during development.



CHAPTER - 6

CIRCUIT DIAGRAM AND CONNECTIONS

1. TCS3200 Color Sensor to Pro Mini:

| TCS3200 Pin | Pro Mini Pin | Description |
|-------------|--------------|------------------------|
| VCC | VCC (5V) | Power supply (5V) |
| GND | GND | Ground |
| OUT | D2 | Frequency output |
| S0 | D3 | Frequency scaling pin |
| S1 | D4 | Frequency scaling pin |
| S2 | D5 | Color filter selection |
| S3 | D6 | Color filter selection |

2. 0.96" OLED I2C Display (4-pin) to Pro Mini:

| OLED Pin | Pro Mini Pin | Description |
|----------|-------------------|--------------------------|
| VCC | VCC (3.3V or 5V)* | Power (check OLED specs) |
| GND | GND | Ground |
| SDA | A4 | I2C Data |
| SCL | A5 | I2C Clock |

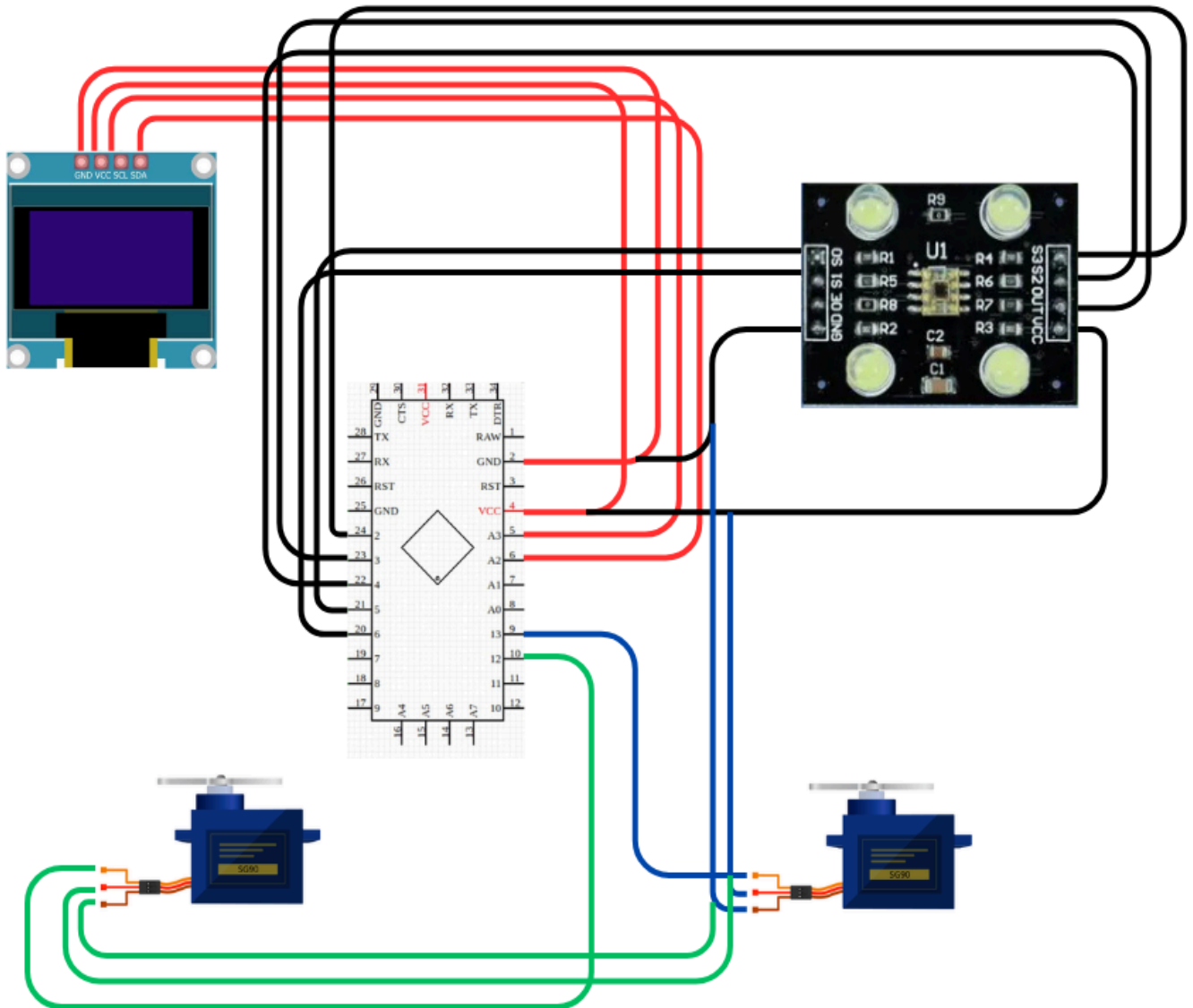
3. Servo Motor 1 and 2 to Pro Mini:

| Servo Motor | Signal Pin | Pro Mini Pin | Description |
|-------------|-------------|--------------|---------------|
| Servo 1 | Yellow | D9 | Control pin |
| Servo 2 | Yellow | D10 | Control pin |
| Both Servo | Red | 5V External* | Power supply |
| Both Servo | Brown/Black | GND | Common Ground |

4. FTDI USB to Serial Adapter (for uploading code):

| FTDI Pin | Pro Mini Pin |
|----------|--------------|
| VCC | VCC |
| GND | GND |
| TX | RX (D0) |
| RX | TX (D1) |
| DTR | DTR |

CIRCUIT DIAGRAM



CHAPTER - 7

PROGRAM:

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Servo.h>

Servo servo1; // Positioning servo
Servo servo2; // Gate servo

// OLED setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// TCS3200 pin definitions
const int s0 = 9;
const int s1 = 8;
const int s2 = 7;
const int s3 = 6;
const int out = 5;

int red, green, blue;
String detectedColor = "";

void setup() {

  serial.begin(9600);

  servo1.attach(3);
  servo2.attach(4);
  servo2.write(0); // Keep gate initially closed

  // Set TCS3200 pins
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
```

```

digitalWrite(s0, HIGH);
digitalWrite(s1, HIGH);

// OLED init
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println(F("OLED not found"));
  while (true);
}

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 0);
display.println("Color Sensor Init");
display.display();
delay(1000);
}

void loop() {
  readColor();
  detectColor();

  Serial.print("R: "); Serial.print(red);
  Serial.print(" G: "); Serial.print(green);
  Serial.print(" B: "); Serial.println(blue);

  // Display on OLED
  display.clearDisplay();
  display.setCursor(0, 0);
  display.print("R: "); display.println(red);
  display.print("G: "); display.println(green);
  display.print("B: "); display.println(blue);
  display.print("Color: "); display.println(detectedColor);
  display.display();

  delay(1000);
}

void readColor() {
  digitalWrite(s2, LOW); digitalWrite(s3, LOW);
  delay(100);
  red = pulseIn(out, LOW);

  digitalWrite(s2, HIGH); digitalWrite(s3, HIGH);
  delay(100);
  green = pulseIn(out, LOW);

  digitalWrite(s2, LOW); digitalWrite(s3, HIGH);

```

```

    delay(100);
    blue = pulseIn(out, LOW);
}

void detectColor() {
    // Match RGB ranges to known colors
    if (inRange(red, 3, 10) && inRange(green, 8, 13) && inRange(blue, 13, 18)) {
        detectedColor = "YELLOW";
    } else if (inRange(red, 5, 10) && inRange(green, 18, 23) && inRange(blue, 18,
23)) {
        detectedColor = "ORANGE";
    } else if (inRange(red, 8, 13) && inRange(green, 18, 23) && inRange(blue, 10,
15)) {
        detectedColor = "ROSE";
    } else if (inRange(red, 65, 100) && inRange(green, 60, 75) && inRange(blue,
50, 70)) {
        detectedColor = "BLACK";
    } else if (inRange(red, 5, 10) && inRange(green, 5, 10) && inRange(blue, 5, 10))
    {
        detectedColor = "WHITE";
    } else if (inRange(red, 0, 40) && inRange(green, 40, 100) && inRange(blue, 40,
100)) {
        detectedColor = "RED";
    } else if (inRange(red, 40, 100) && inRange(green, 40, 100) && inRange(blue,
0, 40)) {
        detectedColor = "BLUE";
    } else if (inRange(red, 40, 100) && inRange(green, 0, 40) && inRange(blue, 40,
100)) {
        detectedColor = "GREEN";
    } else if (inRange(red, 18, 23) && inRange(green, 30, 35) && inRange(blue, 19,
23)) {
        detectedColor = "MAGENTA";
    } else {
        detectedColor = "UNKNOWN";
    }
}

bool inRange(int val, int minVal, int maxVal) {
    return val >= minVal && val <= maxVal;
}

// Move servo1 to bin based on detectedColor
if (detectedColor == "RED") {
    servo1.write(0);
} else if (detectedColor == "GREEN") {
    servo1.write(30);
} else if (detectedColor == "BLUE") {
    servo1.write(60);
} else if (detectedColor == "YELLOW") {

```

```
servo1.write(90);
} else if (detectedColor == "BLACK") {
    servo1.write(120);
} else if (detectedColor == "WHITE") {
    servo1.write(150);
} else {
    // Unknown color - no movement
    servo1.write(90); // Neutral
}

delay(500); // Allow time for servo1 to rotate

// Open gate with servo2
servo2.write(90); // Open
delay(1000);      // Wait for ball to drop
servo2.write(0);  // Close gate
```

CHAPTER - 8

8.1 IMPLEMENTATION:

The project is designed to automatically sort products based on their color using a low-cost embedded system. It involves integrating the following components:

- TCS3200 color sensor: To detect the color of the object.
- Pro Mini ATmega328P: Acts as the central controller.
- OLED Display (I2C 0.96"): To show the detected color in real time.
- Two Servo Motors: To push/sort the object into the correct bin based on its color.

The components are connected as per the circuit described earlier. The Pro Mini is programmed using the Arduino IDE via an FTDI USB-to-Serial adapter. The color sensor is calibrated to identify five specific colors: Red, Green, Yellow, Black, and White.

Once the color is detected, it is displayed on the OLED. Then, a corresponding servo motor is activated to move the product into the correct bin.

8.2 WORKING:

- **Color Detection:**

The TCS3200 sensor detects the color of the object placed beneath it. It uses an array of photodiodes filtered for red, green, and blue, and converts light intensity to frequency, which is read by the Pro Mini.

- **Color Processing:**

The Pro Mini reads the frequency outputs and determines the RGB intensity. Based on predefined thresholds, it identifies the closest matching color (from the five supported colors).

- **Display Output:**

The identified color is displayed in real-time on the OLED I2C screen (e.g., "Color: RED").

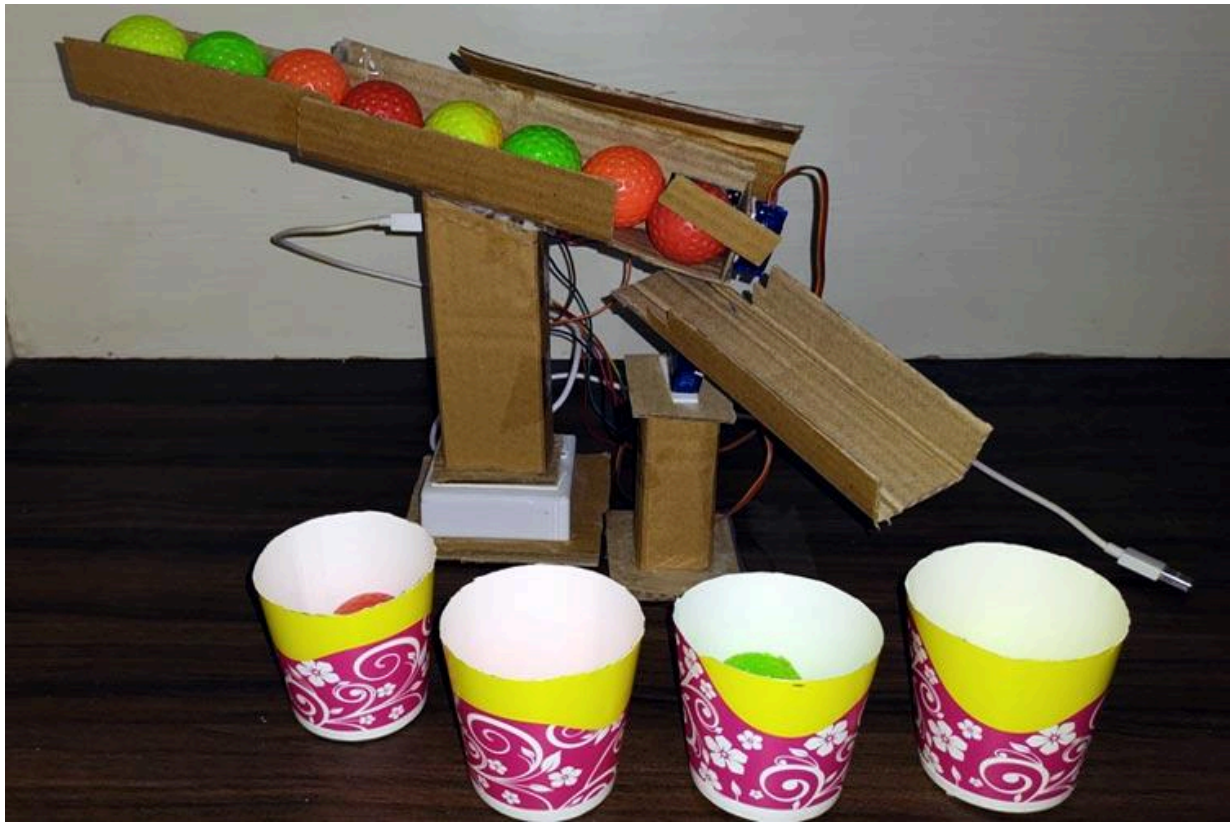
- **Object Sorting:**

Based on the detected color, the Pro Mini sends a PWM signal to the corresponding servo motor. The servo rotates (e.g., 45° or 90°) to push the object into its designated bin.

- **Reset:**

After sorting, the servo returns to its initial position and the system waits for the next object.

8.3 PROTOTYPE/WORKING MODEL:



CHAPTER - 9

9.1 APPLICATIONS:

This system can be implemented in several real-world applications:

- **Agricultural Industry:** For sorting fruits and vegetables based on their ripeness or color.
- **Recycling Plants:** To separate materials like plastics or papers based on color.
- **Food Processing Units:** For quality control by detecting discolored or defective products.
- **Industrial Manufacturing:** To sort components or items based on their visual properties.
- **Retail & Warehousing:** For automated sorting of products or packaging in supply chains.

9.2 LIMITATIONS:

Some of the current limitations of this project include:

- **Limited Color Range:** Can only accurately detect and differentiate five specific colors.
- **Lighting Dependency:** Detection accuracy is affected by ambient light variations.
- **Fixed Sorting Angles:** Limited by predefined servo motor angles.
- **Non-Shape-Aware:** Only color is considered; object shape/size is ignored.
- **Manual Placement:** Objects must be manually positioned under the sensor for detection.

9.3 FUTURE SCOPE:

There is significant potential to enhance this project further:

- Integration with Conveyor Belt: Automate object placement and movement.
- Use of Camera & Image Processing: For detecting complex colors or mixed-color patterns.
- Wireless Connectivity: For remote monitoring and control using Bluetooth/Wi-Fi.
- AI-Based Classification: For more accurate color and object identification using ML.
- Extended Color Library: Add support for more colors and gradients.

CHAPTER - 10

10.1 CONCLUSION:

The project "**Color-Based Product Sorting Using Pro Mini ATmega328P, TCS3200 Color Sensor, OLED Display, and Servo Motors**" presents a **low-cost, efficient, and compact solution** for automating color-based sorting tasks. It uses a color sensor for detection, an OLED for real-time display, and servo motors for directing objects into appropriate bins based on colors like **red, green, yellow, black, and white**. This setup improves speed and accuracy, reducing human labor in sectors such as **small-scale industries and agriculture**. While it currently has limitations like fixed color detection and sensitivity to lighting, it serves as a strong foundation for future enhancements including **AI integration and conveyor automation**.

10.2 REFERENCES:

- Servo Motor Basics – https://www.electronics-tutorials.ws/io/io_7.html
- OLED I2C Display with Arduino Tutorial – <https://randomnerdtutorials.com/>
- Embedded Systems Design – Raj Kamal
- Practical Electronics for Inventors – Paul Scherz and Simon Monk
- TCS3200 Color Sensor Datasheet – Texas Instruments
- Arduino Official Documentation – <https://www.arduino.cc/>