PROJECT REPORT

COLOUR BASED PRODUCT SORTING

Submitted By

YOGESHWARAN.A (513121106709)

VIGNESH K. (513121106710)

TAMIZHARASAN.K (513121106711)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY

VELLORE-2

CONTEXT

ABSTRACT:

This project presents a solution for automating the sorting of products based on their colour using Internet of Things (IoT) technology. Manual sorting of products by colour can be challenging and time-consuming, especially for a large number of items.

The proposed system utilizes an Arduino microcontroller board integrated with a TCS3200 colour sensor and a load cell to detect the colour and weight of the products, respectively. A LCD display is employed to present the product name along with its weight.

By analyzing the colour values in RGB and weight data, the system efficiently sorts the products, reducing manual effort and time. The goal of this project is to develop a cost-effective and reliable device for product sorting to enhance efficiency and productivity in industries.

The implementation of this IoT-based solution offers a practical approach to streamline sorting processes, ultimately leading to time savings and increased operational efficiency.

COMPONENTS USED:

1. ARDUINO UNO R3:

The Arduino Uno R3 is a popular microcontroller board widely used for various electronic projects and prototyping. It features an ATmega328P microcontroller, digital and analog input/output pins, USB connectivity, and power jack. The Uno R3 is user-friendly, versatile, and supports a wide range of sensors and modules, making it ideal for beginners and experienced makers alike.

2. TCS3200 COLOUR SENSOR:

The TCS3200 is a colour sensor module that can detect and measure the intensity of red, green, blue, and clear (no colour) light. It consists of an array of photodiodes with colour filters, a light-to-frequency converter, and an analog-to-digital converter. The sensor can be programmed to output frequency signals corresponding to the intensity of each colour detected. It's commonly used in applications such as colour recognition, sorting, and calibration in various industries, including robotics, automation, and electronics.

3. LCD DISPLAY:

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

4. I2C CONVERTER:

An I2C converter in an LCD display module acts as an intermediary between the display and the microcontroller, such as Arduino. It enables communication using the I2C protocol, reducing the number of pins required for connection. Typically integrated as an I2C backpack, it streamlines wiring and simplifies interfacing. This converter translates data and control signals from parallel to serial, facilitating easier integration into projects. Its inclusion enhances the versatility and ease of use of the LCD display module.

5. SERVO MOTOR:

A servo motor is a self-contained sophisticated electromechanical device that rotates parts of a machine or robot with high efficiency and great accuracy. A servo motor can move slowly and at the same time deliver large torque with great precision and accuracy. This is the reason why it is utilized in robot design, industrial automation, control surface positioning in remote control vehicles, etc.

6. JUMPER WIRES:

Jumper wires are flexible wires with connectors at each end, commonly used in electronics projects to create electrical connections between components on breadboards, modules, or circuit boards. They come in various lengths and colours, facilitating easy identification and organization. Jumper wires enable rapid prototyping and experimentation by allowing components to be quickly interconnected without soldering, making them an essential tool for hobbyists, students, and professionals in electronics and engineering fields.

7. SERVO MOTOR:

A servo motor is defined as an electric motor that allows for precise control of angular or linear position, speed, and torque. It consists of a suitable motor coupled to a sensor for position feedback and a controller that regulates the motor's movement according to a desired setpoint.

8. FOAM BOARDS:

Foam board or foam core board is a rigid, lightweight material that can be molded or fabricated into shapes and painted**.** It is commonly used in architecture, design, costumes and props, mounting prints, displays, and picture frame backing. Foam board consists of three layers that can easily be cut with a knife. It is also used as insulation in the form of rigid board made of foam products that can include polyurethane, polystyrene, and polyisocyanurate.

METHODOLOGY:

1. **Requirement Analysis:**

- Define the scope of product sorting, considering the variety of products and their colour range.
- Assess the volume of products to be sorted to determine the system's throughput capacity.
- Identify any specific industry standards or regulations that need to be adhered to.

2. **Component Selection:**

- Choose Arduino UNO for its versatility and compatibility with various sensors and displays.
- Opt for TCS3200 Colour Sensor due to its accuracy in detecting a wide range of colours.
- Select an LCD i2c display for its simplicity in interfacing with Arduino and clear display output.

3. **Hardware Setup:**

- Connect Arduino UNO to the TCS3200 Colour Sensor and LCD i2c display using appropriate jumper wires.
- Ensure proper positioning of the colour sensor for optimal colour detection from the products.
- Double-check the wiring connections to prevent any hardware malfunction during operation.

4. **Programming Arduino:**

- Develop code to initialize the colour sensor and LCD display modules.
- Implement algorithms to interpret RGB colour values and map them to corresponding product categories.
- Include error handling mechanisms to manage unexpected scenarios during sorting operations.

5. **Data Acquisition and Calibration:**

- Collect sample data of product colours and their corresponding RGB values for calibration.
- Fine-tune the system's colour detection thresholds based on collected data to improve accuracy.
- Validate the sorting results against a reference dataset to ensure consistency and reliability.

6. **Testing and Validation:**

- Conduct rigorous testing using a diverse set of products with different colours and shapes.
- Verify the sorting accuracy by comparing the system's results with manual sorting outcomes.
- Address any discrepancies or errors encountered during testing and refine the system accordingly.

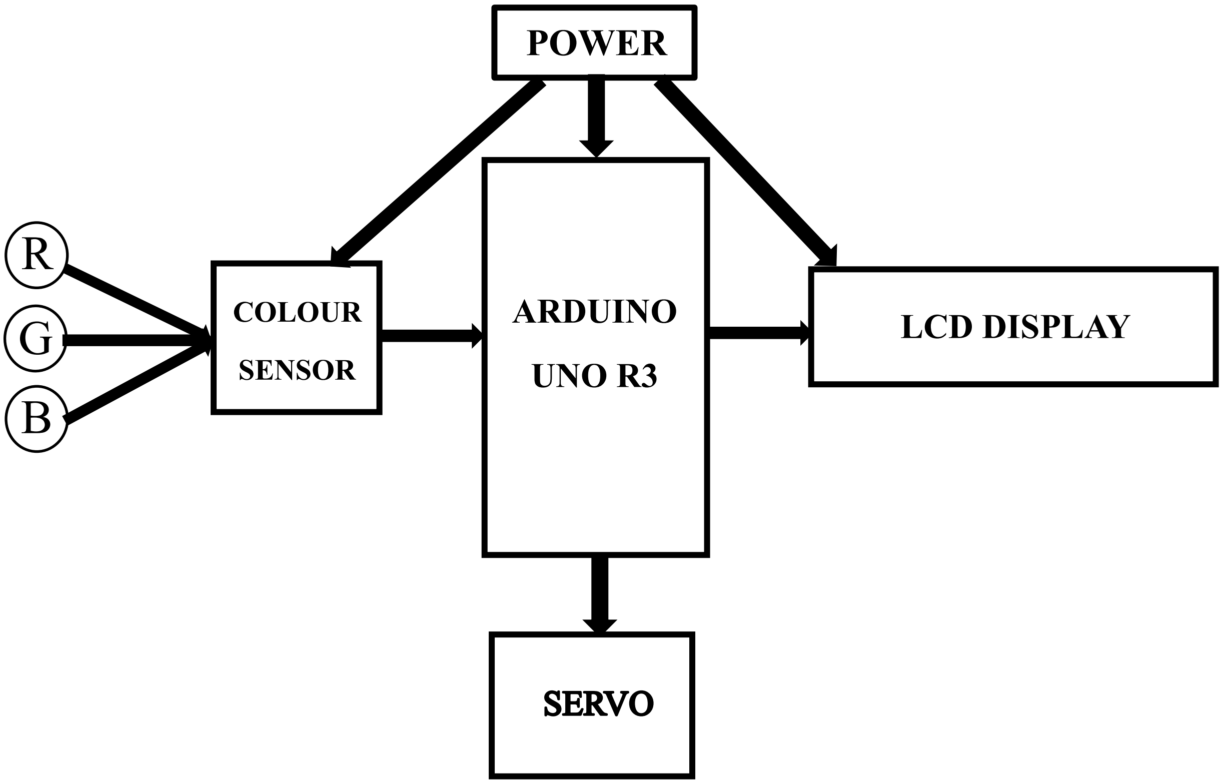
7. **Integration with IoT Platform (Optional):**

- Integrate the sorting system with an IoT platform for remote monitoring and management capabilities.
- Enable features such as real-time alerts, data analytics, and remote configuration updates.
- Ensure seamless integration between the IoT platform and the hardware components for efficient operation.

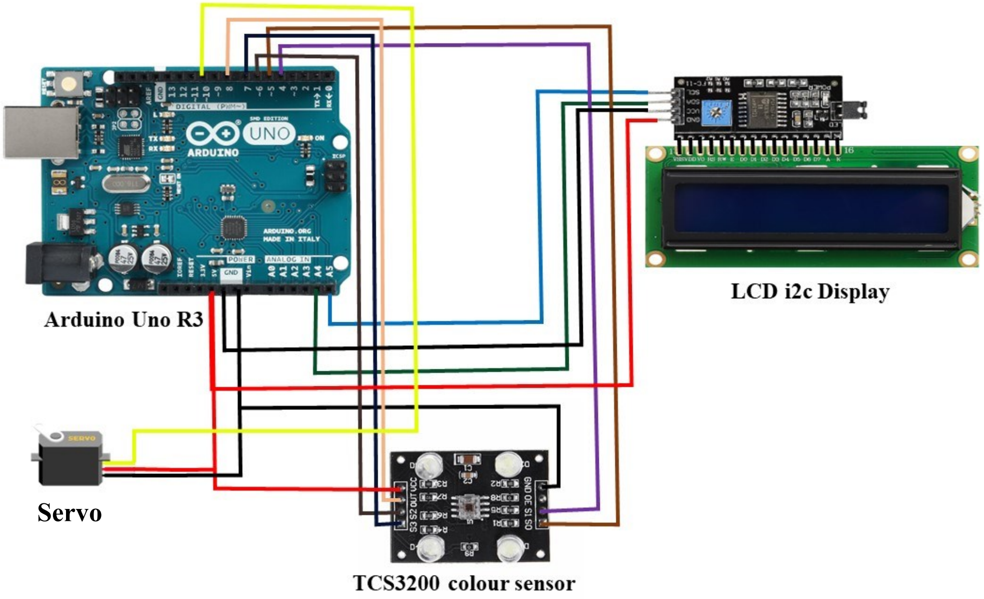
8. **Deployment and Optimization:**

- Deploy the sorting system in the production environment after thorough testing and validation.
- Monitor system performance and gather feedback from operators for optimization opportunities.
- Continuously refine the system through software updates and hardware enhancements to improve efficiency and reliability.

BLOCK DIAGRAM:



CIRCUIT DIAGRAM:



CODE:

SOFTWARE TOOL: ARDUINO IDE

PROGRAMMING LANGUAGE: C++

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
//I2C pins declaration
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define S0_PIN 5
#define S1_PIN 4
#define S2_PIN 7
#define S3_PIN 6
#define OUT_PIN  8
Servo Servo;
void setup()
{
lcd.init();
lcd.backlight();
Servo.attach(10);
// Set the S0, S1, S2, S3 Pins as Output
pinMode(S0_PIN, OUTPUT);
pinMode(S1_PIN, OUTPUT);
pinMode(S2_PIN, OUTPUT);
pinMode(S3_PIN, OUTPUT);
//Set OUT_PIN as Input
pinMode(OUT_PIN, INPUT);
 // Set Pulse Width scaling to 20%
digitalWrite(S0_PIN, HIGH);
digitalWrite(S1_PIN, LOW);
```

```
// EnablE UART for Debugging
Serial.begin(9600);
}
void loop()
{
  int r, g, b;
 r = process_red_value();
 delay(200);
 g = process_green_value();
 delay(200);
 b = process_blue_value();
 delay(200);

 Serial.print("r = ");
 Serial.print(r);
 Serial.print(" ");
 Serial.print("g = ");
 Serial.print(g);
 Serial.print(" ");
 Serial.print("b = ");
 Serial.print(b);
 Serial.print(" ");
 Serial.println();
 if (b>400 && r>400 && g<650)
 {
 Serial.println("green");
 lcd.setCursor(0,0);
 lcd.print(" Product Colour ");
 lcd.setCursor(0,1);
 lcd.print("     Green       ");
```

```
Servo.write(180);
delay(100);
}
else if (r<=463 && b>500 && g>700)
{
Serial.println("orange");
lcd.setCursor(0,0);
lcd.print(" Product Colour ");
lcd.setCursor(0,1);
lcd.print("     Orange     ");
Servo.write(15);
delay(100);
}
else if (r>500 && b<460)
{
Serial.println("Blue");
lcd.setCursor(0,0);
lcd.print(" Product Colour ");
lcd.setCursor(0,1);
lcd.print("      Blue      ");
Servo.write(135);
delay(100);
}
else if (r<400)
{
 Serial.println("yellow");
 lcd.setCursor(0,0);
 lcd.print(" Product Colour ");
 lcd.setCursor(0,1);
 lcd.print("    Yellow     ");
 Servo.write(45);
 delay(100);
 }
else if (820<g && r<600 && b<610 )
{
Serial.println("Brown ");
lcd.setCursor(0,0);
lcd.print(" Product Colour ");
lcd.setCursor(0,1);
lcd.print("     Brown       ");
Servo.write(90);
delay(100);
}
else{
lcd.setCursor(0,0);
lcd.print("    Place the     ");
lcd.setCursor(0,1);
lcd.print("     Product      ");
Servo.write(90);
delay(1000);
}
}
int process_red_value()
{
digitalWrite(S2_PIN, LOW);
digitalWrite(S3_PIN, LOW);
int pulse_length = pulseIn(OUT_PIN, LOW);
return pulse_length;
}
```

```
int process_green_value()
{
digitalWrite(S2_PIN, HIGH);
digitalWrite(S3_PIN, HIGH);
int pulse_length = pulseIn(OUT_PIN, LOW);
return pulse_length;
}
int process_blue_value()
{
digitalWrite(S2_PIN, LOW);
digitalWrite(S3_PIN, HIGH);
int pulse_length = pulseIn(OUT_PIN, LOW);
return pulse_length;
}
```

WORKING:

1. **Setup:**

   Connect the TCS3200 colour sensor and the LCD i2c display to the Arduino UNO board using jumper wires, ensuring proper wiring and connections.

2. **Calibration:**

   Calibrate the colour sensor to accurately detect and differentiate between different colours. This may involve adjusting parameters such as sensitivity and integration time to optimize colour detection.

3. **Data Acquisition:**

   Place the products to be sorted in front of the colour sensor. The sensor detects the colour of each product and converts it into corresponding RGB values.

4. **Colour Analysis:**

   The Arduino UNO reads the RGB values from the colour sensor and compares them with pre-defined RGB values stored in its memory for different product colours.

5. **Product Identification:**

   Based on the closest match between the detected colour and the pre-defined colour values, the Arduino UNO determines the corresponding product and retrieves its name from a database.

6. **Display Output:**

   The name of the identified product is displayed on the LCD i2c display, indicating the successful sorting of the product based on its colour.

7. **Sorting Process:**

   Depending on the application, the sorted products can be directed to different bins or conveyor belts for further processing or packaging based on their identified colours.

8. **Repeat:**

   The system continues to scan and sort products in real-time, providing a continuous and automated sorting process based on colour.

CONCLUSION:

In conclusion, the development of our colour-based product sorting system represents a significant step forward in industrial automation, offering a practical and efficient solution to the challenges associated with manual sorting processes. By leveraging Internet of Things (IoT) technology, we have created a cost-effective and reliable device capable of swiftly and accurately sorting products based on their colours.

Through the integration of Arduino UNO, TCS3200 colour sensor, and LCD display, our system provides a streamlined approach to product sorting, reducing both time and labor requirements. By analyzing RGB values and employing a database of pre-defined colour parameters, our device accurately identifies and displays the names of sorted products in real-time.

The implementation of this project holds several key advantages. It enhances operational efficiency by automating a previously manual task, thereby increasing productivity and reducing costs for industries handling large volumes of products. Furthermore, the system's versatility and scalability make it suitable for a wide range of applications, from manufacturing and logistics to retail and beyond.

Moving forward, we envision further enhancements and refinements to our system, including integration with IoT platforms for remote monitoring and management capabilities. Additionally, ongoing optimization efforts will focus on improving sorting accuracy, expanding the range of detectable colours, and enhancing user interface features for seamless operation.

Overall, our colour-based product sorting project underscores the transformative potential of IoT technology in revolutionizing traditional industrial processes, offering a glimpse into a future where automation and efficiency go hand in hand to drive progress and innovation.