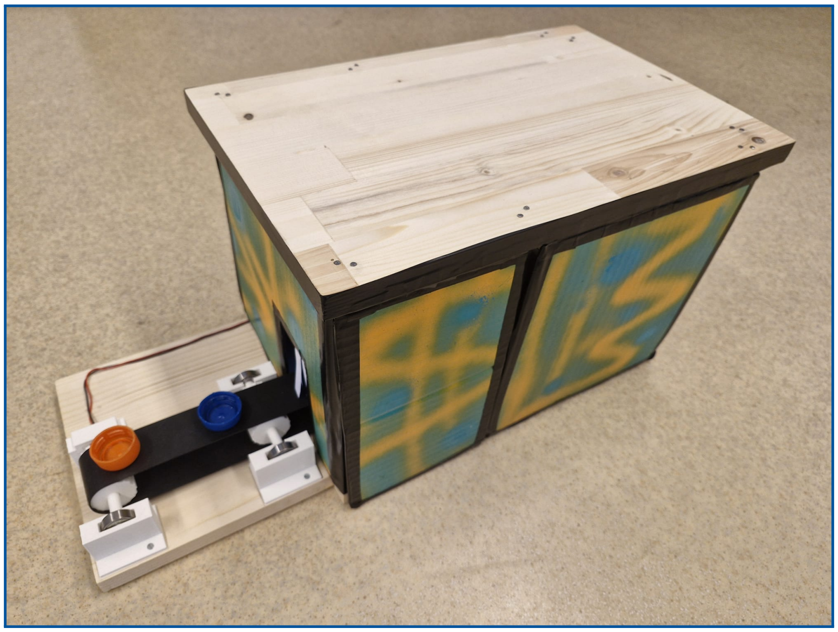
Dual Color Sorter



Abstract

This project presents an automated object sorting system implemented with a conveyor belt. The system is designed to identify and sort objects based on their color (orange or blue) using a combination of an ultrasonic distance sensor and a color sensor. A paddle mechanism redirects objects to appropriate paths based on their detected color. This document outlines the design, implementation, and testing of the system, along with its potential applications in industrial automation.

Authors:
Badea Cătălin-Gabriel, Matei David, Atudosiei Andrei Cristian

Date: May 2025

Contents

1 Domain Overview **3**
1.1 Introduction . . . 3
1.2 The Importance of Sorting Systems . . . 3
1.3 Similar Existing Solutions . . . 3
1.4 Our Approach . . . 4

2 Project Objectives **5**

3 Implementation **6**
3.1 General Description . . . 6
3.2 System Components . . . 6
3.3 Electrical Circuit Design . . . 8
3.4 Technical Specifications . . . 9
3.5 Programming and Microcontroller Code . . . 10

4 Testing and Calibration **11**
4.1 TCS230 Color Detector . . . 11
4.1.1 Calibration Process Structure . . . 11
4.1.2 Setting Color Thresholds . . . 11
4.1.3 Color Distance Calculation . . . 11
4.1.4 Calibration Completion . . . 11
4.2 HC-SR04 Ultrasonic Sensor . . . 11
4.3 DC Motors . . . 12
4.3.1 Basic DC Motor Test . . . 12
4.3.2 DC Motor with Encoder (CGM12-N20VA-8200E) . . . 12

5 Problems Encountered **12**

6 Potential Improvements and Future Extensions **13**

7 Conclusion **14**
7.1 Individual Contributions . . . 14
7.2 Bill Of Materials . . . 15

1 Domain Overview

1.1 Introduction

The domain of this project is firmly situated within **intelligent transportation systems (ITS)**, with a specific focus on automated object classification and routing. This system demonstrates the principles of smart logistics by using a conveyor belt to sort objects based on their color—specifically identifying and separating orange and blue items.

By integrating basic color-based classification, the system mimics intelligent transport technologies where real-time decisions are made based on sensor input. This leads to improved accuracy, reduced manual intervention, and increased efficiency in the handling and routing of physical items within broader transportation and logistics frameworks.

1.2 The Importance of Sorting Systems

Sorting systems are fundamental to the efficient handling and management of data, objects, and processes across a wide range of industries. Whether applied in computer science, logistics, manufacturing, or administration, an effective sorting mechanism improves clarity, reduces processing time, and enhances user experience.

By organizing elements based on defined criteria—such as alphabetical order, numerical value, priority, or category—sorting enables faster retrieval, better data analysis, and streamlined operations. In large-scale systems, such as databases or automated warehouses, sorting becomes critical for performance optimization and scalability.

Moreover, it contributes to error reduction and supports automation by establishing a predictable structure. As the amount of information and complexity of workflows continue to grow, the role of sorting systems becomes increasingly vital for ensuring order, efficiency, and informed decision-making.

1.3 Similar Existing Solutions

Modern industrial sorting systems increasingly rely on advanced technologies to improve accuracy, speed, and flexibility. Table 1 highlights some of the most commonly used methods and their applications.

Technology	**Application**
Machine vision	Cameras and image processing for sorting based on color, shape, or size
AI and machine learning	Pattern recognition for complex or variable sorting tasks
RFID technology	Wireless identification and sorting of tagged items

Table 1: Advanced technologies in industrial sorting systems

These technologies are widely used in industries such as manufacturing, recycling, and logistics. Machine vision handles visual analysis, AI enables smarter decisions based on learned patterns, and RFID offers efficient tracking and categorization without visual input.

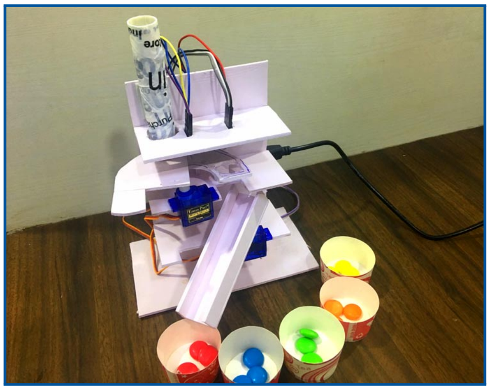


Figure 1: Multiple Color Sorter System Machine

Figure 2: Apple Sorter with Mechanical Redirectors

1.4 Our Approach

Our approach is a simplified prototype inspired by industrial sorting systems. While commercial systems rely on advanced technologies, our focus is on color-based sorting using basic sensors and mechanical actuation.

Created as a didactic prototype, this system aims to teach and experiment with the core principles of automated sorting. Using an ultrasonic sensor for detection and a color sensor for classification, objects are sorted based on their color, with a DC motor controlled paddle redirecting them accordingly.

Though simpler than industrial solutions, our approach serves as a foundation for learning and exploring automated sorting concepts. Furthermore, the colored objects used in this project can simulate containers or casings for items of interest, such as pills, electronic components, or food products. This abstraction allows the system to be relevant in various practical scenarios, including pharmaceutical packaging, component assembly lines, or quality control in food processing. However, the current paddle-based mechanism might be better replaced with a gentler alternative to avoid damaging sensitive items.

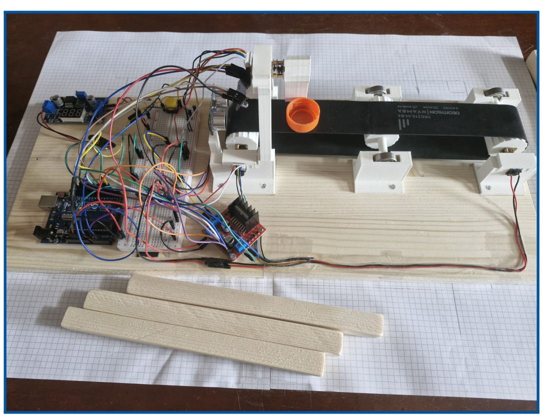


Figure 3: Wiring Closer View

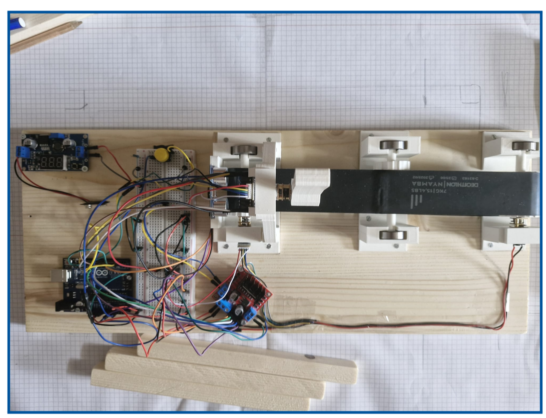


Figure 4: Wire Top View

2 Project Objectives

The development of our automated object sorting system is guided by the following key objectives:

Objective 1: Conveyor-Based Transport System Development

- Design and implement a reliable conveyor belt mechanism
- Ensure smooth and consistent object movement
- Configure appropriate speed for accurate detection and sorting

Objective 2: Precise Color Detection Implementation

- Integrate the TCS230 color sensor for accurate color identification
- Develop algorithms to differentiate between orange and blue objects
- Calibrate the sensor for optimal performance under varying lighting conditions

Objective 3: Automated Object Redirection Mechanism

- Design an efficient paddle actuation system
- Program the rotary motor for precise angular positioning using an encoder
- Ensure reliable sorting with minimal errors

Objective 4: Synchronized Process Control

- Implement the HC-SR04 ultrasonic sensor for accurate object detection
- Create timing sequences for coordinated operation of all system components
- Develop robust control logic for continuous operation

Objective 5: System Optimization and Performance Validation

- Conduct comprehensive testing under operational conditions
- Measure and improve sorting accuracy rates
- Fine-tune timing parameters for maximum throughput

These objectives serve as the foundation for developing a functional, efficient, and reliable color sorting system applicable to small-scale industrial environments where rapid and precise object handling is required.

3 Implementation

3.1 General Description

The project consists of an **automated object sorting system** on a conveyor belt, designed to sort two types of colors, which can be configured depending on the system's calibration. Although the system can be adjusted to detect and sort any two colors, throughout this project we will focus on the case of sorting **orange** and **blue** objects.

The system operates through the following process:

1. Pressing a button activate/deactivates the conveyor belt, which is powered by two motors and moves objects smoothly along its path.

2. Objects reach a detection zone where an *HC-SR04 ultrasonic distance sensor* is positioned.

3. When an object reaches the predetermined distance threshold, the belt automatically stops.

4. The *TCS230 color sensor* identifies its color.

5. Based on the detected color, a paddle operated by another *GA12 N20 6V motor* redirects the object:
   - **Orange** objects are directed to the **left** path
   - **Blue** objects are directed to the **right** path

6. After redirection, the system pauses for approximately **? second** to ensure complete processing.

7. The conveyor belt resumes operation to transport the next object.

This automated cycle continues, enabling efficient and precise sorting of objects by color. The system's design emphasizes simplicity and reliability.

3.2 System Components

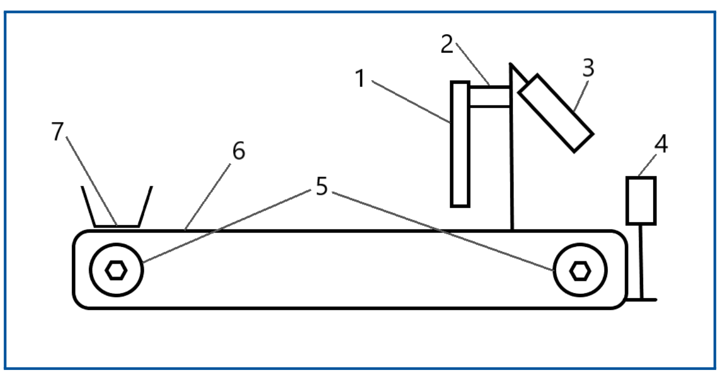


Figure 5: Structure Diagram

Above, a diagram outlines the structure of the project and below there is a list of the components used, along with a table regarding the key system components and their functions.

No.	Component Name
1	Paddle
2	DC-motor (Paddle Controller)
3	Color Sensor
4	Distance Sensor
5	DC-motors (Belt Controllers)
6	Conveyor Belt
7	Object to be sorted

Table 2: List of system components by number and name

Component	Type/Model	Function
Conveyor Belt	Dual-motor driven	Transports objects through the sorting system
Distance Sensor	HC-SR04 Ultrasonic	Detects object presence and stops the conveyor belt in the right position
Color Sensor	TCS230	Identifies the color of objects (orange vs. blue), sending the signal to the microcontroller
Motor Driver	L298N	Dual H-bridge driver for controlling both the conveyor belt and paddle DC motors
Buck Convertor	LM2596	Adjustable step-down (buck) converter that provides a stable 6V supply from the 9V battery
Actuators	3 × GA12-N20 6V DC-motor	Controls the paddle for object redirection as well as the conveyor belt
Control Unit	Arduino UNO Microcontroller	Processes sensor data and controls system components
Power Supply	2 × 9V Battery	Provides electrical power to all system components, including the conveyor belt

Table 3: Key system components and their functions

Note that the box can be opened on any side, except for the front, to provide easy access to the circuits and wires inside. This design allows for convenient maintenance and modifications, ensuring that all components are accessible without disturbing the main structure of the system.

3.3 Electrical Circuit Design

In this section, the electrical circuit design of the sorting system is presented. The system utilizes multiple components that work together to process data and control the actuators for object sorting. Below is a simplified block diagram that shows how these components are interconnected.

At the core of the system is the Arduino microcontroller, which processes data from the sensors and controls the actuators. The ultrasonic distance sensor is connected to the Arduino and is powered with 5V from it. The color sensor is also connected to the Arduino, but it operates at 3.3V. The Arduino controls the motors through an H-bridge. The H-bridge drives two motors that control the conveyor belt simultaneously and one motor for the paddle mechanism, which redirects the objects. The motors are powered with 6V from a buck converter, which steps down the 9V battery supply to 6V (the operating voltage of the motors). The Arduino itself is powered by a separate 9V battery to ensure proper functionality and to prevent the higher current draw from the motor line from damaging the microcontroller.

The block diagram below illustrates the overall flow of power and control signals within the system.

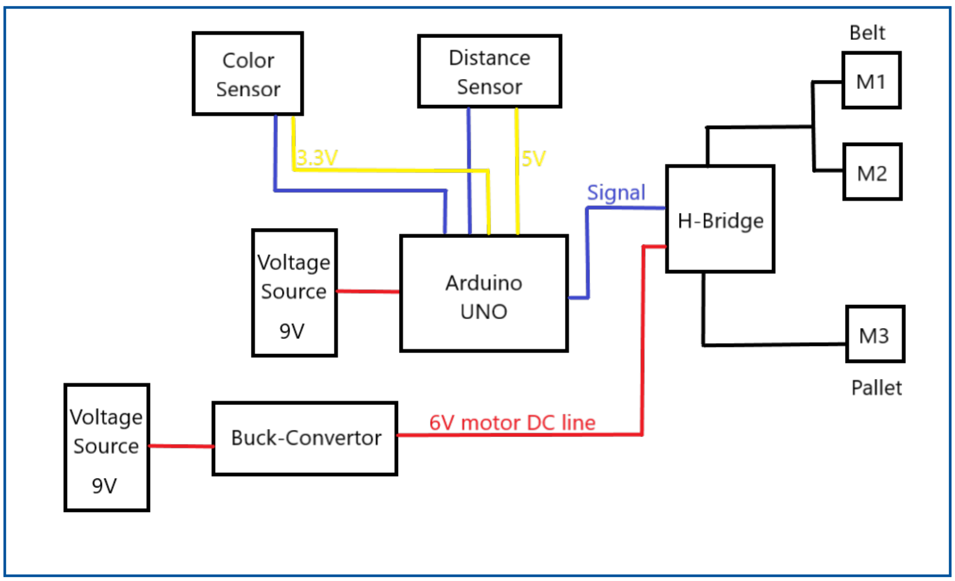


Figure 6: Block diagram of the sorting system

Following this section, the detailed circuit schematic is provided, showing the exact wiring of each component and how they are physically connected to one another.

It is important to note that the circuit includes a button designed to stop the conveyor belt, allowing for debugging or turning off the system when not in use.

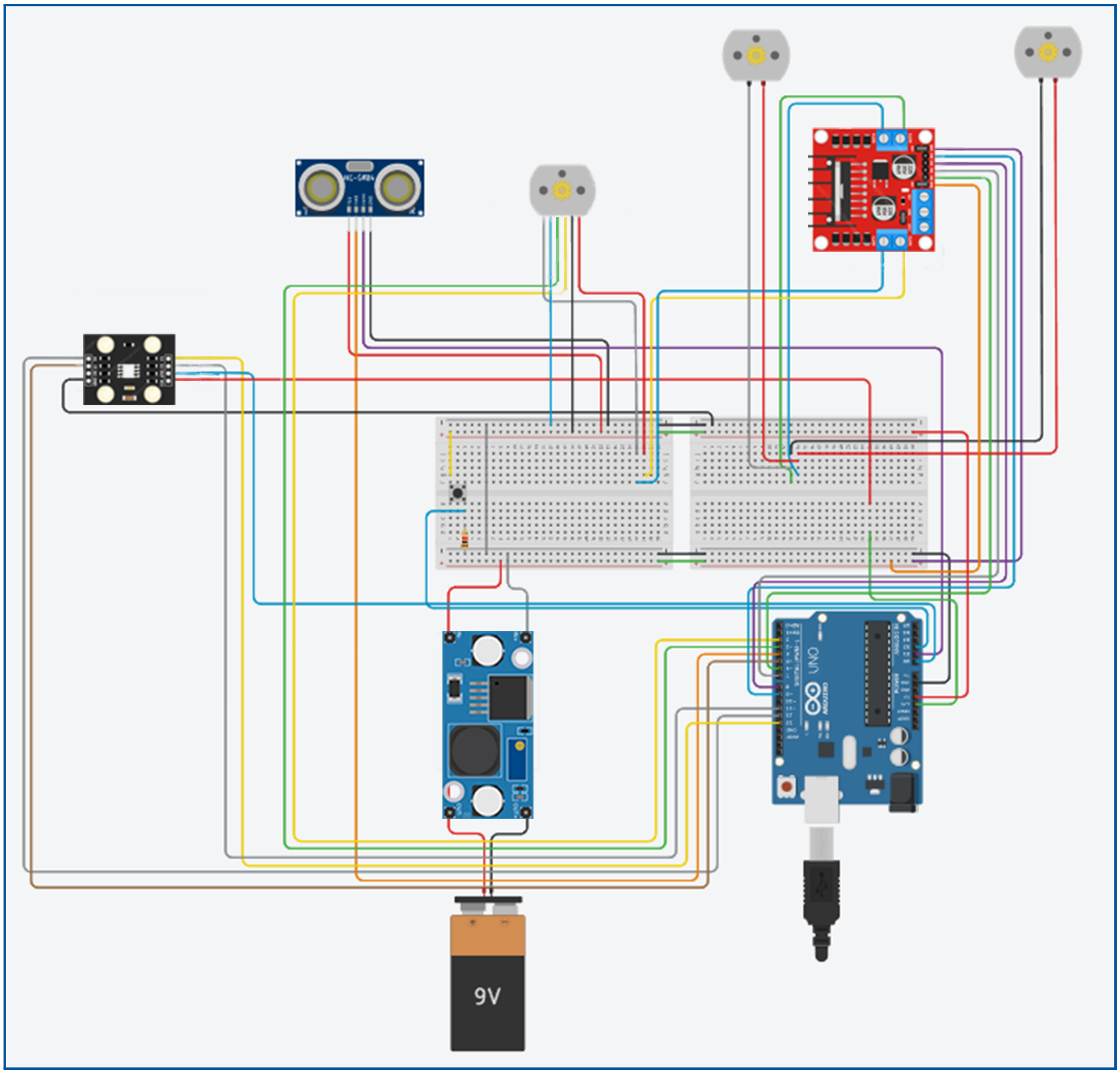


Figure 7: Detailed wiring diagram of the system

3.4 Technical Specifications

Parameter	**Specification**
Conveyor Length/Width/Height	26 cm / 4 cm / 4 cm
Total System Dimensions	54 cm x 25 cm x 24 cm (L x W x H)
Total System Dimensions Needed (open)	75 cm x 68 cm x 24 cm (L x W x H)
Maximum Object Size	3 cm × 3 cm × 4 cm
Power Consumption	35W
Operating Temperature	10-40°C
Power Supply	2 x 9V Battery

Table 4: System technical specifications

3.5 Programming and Microcontroller Code

In this section, the programming logic of the sorting system is presented, focusing on the microcontroller code. The Arduino is responsible for processing the data from the sensors, making decisions based on the input, and controlling the actuators to sort the objects.

The code is written in C++ using the Arduino IDE, and it communicates with the various system components, including the ultrasonic distance sensor, color sensor, and motors.

Algorithm 1: Main logic for the color sorting system

```
setup Initialize pins;
loop Calculate how long the button was pressed;
if pressed time less than 2s then
    if not in calibration mode then
        if system is off then
            Turn on system;
        else
            Turn off system;
        end
    else
        Process calibration step;
    end
else
    Start calibration mode;
end
if system is on then
    Start conveyor belt;
    Calculate distance to object;
    if object reached paddle area then
        Stop conveyor belt;
        Detect color;
        if color is orange then
            Rotate paddle left;
        else
            if color is blue then
                Rotate paddle right;
            else
                Resume conveyor belt;
            end
        end
    end
end
else
    Stop conveyor belt;
end
```

This code is a basic framework for reading sensor data and triggering actions based on specific conditions.

4 Testing and Calibration

4.1 TCS230 Color Detector

4.1.1 Calibration Process Structure

Calibration is divided into four stages: white, black, blue, and orange.

- **Stage 1 (White):** The user places the sensor on a white surface and presses the button. The system reads the Red, Green, and Blue values and stores them in variables. After a short button press, it advances to the next stage.
- **Stage 2 (Black):** The user places the sensor on a black surface and presses the button. The system reads and saves the RGB values. After a short press, it moves on.
- **Stage 3 (Blue):** The user places the sensor on a blue surface and presses the button. The system reads and saves the RGB values. Then proceeds.
- **Stage 4 (Orange):** The user places the sensor on an orange surface and presses the button. The system reads and saves the RGB values. Calibration is then complete.

4.1.2 Setting Color Thresholds

For each color component (Red, Green, Blue), the minimum and maximum thresholds are calculated from the white and black readings:

$$\begin{aligned} \text{Red:} &\quad \texttt{pragRosuMin}, \texttt{pragRosuMax} \\ \text{Green:} &\quad \texttt{pragVerdeMin}, \texttt{pragVerdeMax} \\ \text{Blue:} &\quad \texttt{pragAlbastruMin}, \texttt{pragAlbastruMax} \end{aligned}$$

Values outside these ranges are considered “unknown.”

4.1.3 Color Distance Calculation

During object detection, the system compares the sensed RGB values to the calibrated blue and orange references using:

$$\text{distance} = (r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2$$

If the distance to orange is less than the distance to blue, the color is detected as orange; otherwise it is detected as blue.

4.1.4 Calibration Completion

After all four stages, the flag `senzorCalibrat = true` is set and the user is notified. The system is now ready for normal operation.

4.2 HC-SR04 Ultrasonic Sensor

To test the HC-SR04, a simple Arduino sketch triggers the sensor continuously, measures the echo return time, computes the distance, and prints the results to the serial monitor. Real-time readings allow verification of the sensor's accuracy.

4.3 DC Motors

4.3.1 Basic DC Motor Test

An Arduino sketch was used to spin the motors first in one direction and then in reverse, confirming proper wiring and operation.

4.3.2 DC Motor with Encoder (CGM12-N20VA-8200E)

For precise one-revolution control, the encoder's pulse count per full rotation was calibrated:

Encoder Overview The encoder emits pulses ("ticks") proportional to shaft rotation.

Estimating Pulses per Revolution Based on the motor's gear ratio and encoder spec, an initial pulse estimate was made.

Testing and Adjustment A sketch rotated the motor while counting ticks. After each trial, the pulse count was tweaked until one commanded revolution matched one full turn.

Finalizing Calibration The correct tick count was hard-coded into the program so that each "full revolution" command yields precise rotation.

5 Problems Encountered

During the development of our project, we encountered a series of technical and practical challenges that required iterative problem-solving and adaptation. Below is a summary of the most significant issues and how we addressed them:

- **Color sensor instability:** Initially, the color sensor did not function reliably. We discovered it was extremely sensitive to minor changes in ambient lighting. To address this, we built a small enclosure around it to maintain consistent lighting conditions.
- **Encoder setup difficulties:** It took us some time to properly understand and use the rotary encoder. One of the main issues was that it only worked correctly when connected to pins 2 and 3 of the Arduino, which limited our initial wiring options.
- **Circuit redesign:** We had to completely rebuild the circuit once in order to optimize pin usage. This allowed us to support all the required functionalities more effectively.
- **Power supply decision:** Finding a suitable power supply was a challenge. In the end, we chose to use a battery, as it offered a more compact and portable solution, and it fit the educational purpose of the project.
- **3D printing issues:** The 3D-printed parts did not always match the exact dimensions due to plastic expansion and shrinkage. As a result, we had to reprint several components before achieving the desired fit and functionality.
- **Belt selection:** It took us some time to find a belt that was suitable for our design and functional requirements.

6 Potential Improvements and Future Extensions

While the current prototype demonstrates the core principles of automated color-based sorting, there are several clear avenues to enhance its performance, flexibility and educational value. Key directions for future work include:

- **Advanced Color Detection:** Upgrade from a simple color sensor to a multi-channel spectral sensor (e.g. TCS34725) or integrate a camera + OpenCV pipeline to sort dozens of hues and even shapes.
- **LCD Feedback Display:** Integrate an LCD module to present real-time calibration progress—showing each stage (white, black, blue, orange)— as well as continuously display the detected color along with a running count of how many times each color has been identified.
- **Modular Mechanical Design:** Design interchangeable paddles, adjustable guide rails and expandable conveyor segments to handle different object sizes or multiple lanes in parallel.
- **Power Management:** Move from 9V batteries to rechargeable Li-ion packs or a smart buck/boost converter with voltage/current monitoring and power-saving modes.

For a future version of the system (v2.0), an important improvement would be the addition of a second sorting paddle mounted on a separate support arm. This would enable the system to handle up to four distinct object categories based on color, instead of the current binary classification.

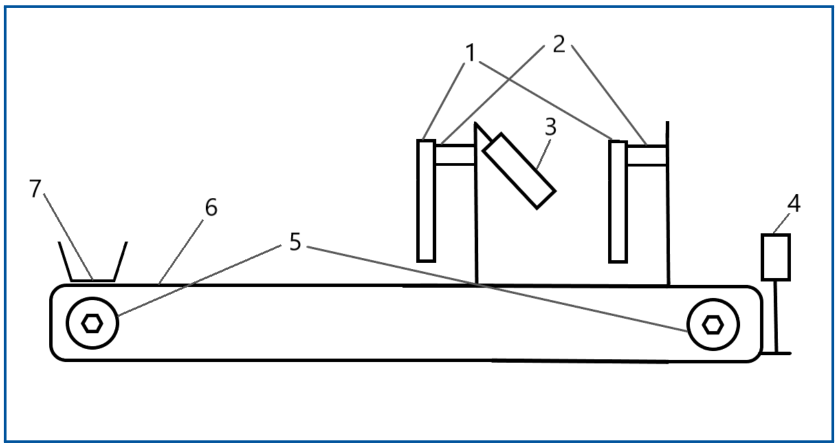


Figure 8: Structure Diagram V2.0

Furthermore, we considered replacing the battery-based power supply with a stable external power source to ensure longer operating times and improved reliability.

However, due to time constraints and prototyping limitations, these enhancements were not implemented in the current iteration.

7 Conclusion

The automated color sorting system successfully demonstrates the application of sensor technology and mechanical actuation. The system reliably sorts orange and blue objects with high accuracy and reasonable throughput.

7.1 Individual Contributions

Badea Cătălin-Gabriel:

- Coordonated the whole project
- Designed the circuit and structure of the machine
- Modeled and 3D-printed the support pieces
- Worked on wiring the circuit
- Worked on building and painting the structure of the machine
- Contributed to documentation

Atudosiei Andrei Cristian:

- Developed the code uploaded on the microprocessor
- Color detection algorithm development
- Sensor calibration
- Performance optimization and debugging
- Contributed to documentation

Matei David:

- Documented the schematic
- Worked on wiring the circuit
- Worked on building and painting the structure of the machine
- Purchased and supplied all the components required
- Contributed to documentation
- Edited the video regarding the project completion

7.2 Bill Of Materials

Listed below are all the components used in the realization and assembly of the automated sorting system, together with their prices and sources of acquisition.

Item	Part Number / Model	Qty	Total (€)	Supplier
1.	Rubber Band	1	3.52	Decathlon
2.	HC-SR04	1	2.93	ElecFreaks
3.	TCS230	1	7.62	TAOS
4.	GA12-N20 with Encoder	2	20.32	Handson Technology
5.	GA12-N20	1	5.11	Handson Technology
6.	Arduino Board A000066	1	11.60	OptimusDigital
7.	Jumper Wires (Male–Male)	18	1.10	OptimusDigital
8.	Jumper Wires (Male–Female)	26	0.72	OptimusDigital
9.	Battery Holder 9V	1	0.28	OptimusDigital
10.	Through-Hole Resistor 470 Ohm	1	0.02	OptimusDigital
11.	Tactile Push-Button	1	0.43	OptimusDigital
12.	Power Cable	1	0.76	XAB3
13.	Bearing 608-2Z Welt	4	4.00	Herasib
14.	High Performance Battery 9V	2	4.48	Auchan
15.	L298N - H-Bridge	1	2.72	Optimus Digital
16.	LM2596 - Buck Convertor	1	1.00	Texas Instruments
17.	Verbatim PLA 1.75 mm (750 g)	1	25.00	Verbatim
18.	Plywood Base	1	3.92	Leroy Merlin
19.	Plywood Poles	1	3.92	Leroy Merlin
20.	Plywood Roof	1	4.9	Leroy Merlin
19.	Nails	32	0.82	Leroy Merlin
19.	Screws	21	1.008	Leroy Merlin

Table 5: Bill of Materials

Total Price: 90.048 €

References

[1] **Electronic Principles (Malvino)** – https://www.talkingelectronics.com/Download/Malvino_Electronic-Principles.pdf

[2] **Ultrasonic Sensor HC-SR04 and Arduino – Complete Guide** – https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

[3] **Good Year Industrial Conveyor Belt** - https://goodyearbelts.com/wp-content/uploads/2024/02/New-Industrial-Catalog-2024-FINAL.pdf

[4] **Contitech Lightweight Conveyor Belt** https://www.goodyearbelting.com/pdf/Contitech_2015_LW_Conv_Belt_Catalog/pdf/Contitech_2015_LW_Conv_Belt_Catalog.pdf

---

Technical datasheets for every component used in this project are listed below:

[5] **HC-SR04** – https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[6] **TCS230** – https://www.mouser.com/datasheet/2/588/cs230-e33-1214740.pdf

[7] **GA12-N20 6V DC Motor** – https://www.handsontec.com/dataspecs/GA12-N20.pdf

[8] **Arduino Uno Microcontroller** – https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[9] **9V Battery (Energizer 522)** – https://data.energizer.com/pdfs/522.pdf

[10] **Conveyor Belt Catalog (Goodyear, 2024)** – https://goodyearrubberproducts.lat/wp-content/uploads/2024-GRP-Conveyor-Belt-Catalog.pdf

[11] **L298N Module** – https://www.alldatasheet.com/datasheet-pdf/view/22440/STMICROELECTRONICS/L298N.html

[12] **LM2596 Module** – https://www.ti.com/lit/ds/symlink/lm2596.pdf