Journal of Electrical Engineering, Electronics, Control and Computer Science – JEEECCS, Volume 11, Issue 38, pages 1-8, 2025

Design and Implementation of an Arduino-Based Automated Color Sorting System

1st Dinică Răzvan-Andrei
Faculty of Electronics, Communications and Computers
National University of Science and Technology Politehnica Bucharest
Pitesti, Romania
razvanandreidinica@gmail.com

3rd Drăguşin Sebastian-Alexandru
Department of Electronics, Computers and Electrical Engineering
National University of Science and Technology Politehnica Bucharest
Pitesti, Romania
dragusin.sebi@yahoo.com

2nd Stanciu Dumitru-Cătalin
Faculty of Electronics, Communications and Computers
National University of Science and Technology Politehnica Bucharest
Pitesti, Romania
cstanciuzedx01c@gmail.com

4th Bizon Nicu
Department of Electronics, Computers and Electrical Engineering
National University of Science and Technology Politehnica Bucharest
Pitesti, Romania
nicubizon@yahoo.com

Abstract –This paper presents the design and implementation of a low-cost, Arduino-based automated color sorting system intended for educational and small-scale industrial applications. The architecture integrates a TCS3200 color sensor for spectral detection, an SG90 servo actuator for diverting items, a 1602 LCD (Liquid Crystal Display), for local feedback, and an acoustic buzzer for status indication, all orchestrated by an Arduino microcontroller. The sensing chain converts reflected RGB (Red, Green, and Blue) intensities to frequency, which are sampled and mapped to calibrated color classes; a rule-based decision layer then commands the sorting actuator. The system emphasizes modularity, reproducibility, and ease of calibration, detailing hardware interfacing, embedded firmware structure (setup/loop pattern), and version-controlled development workflows. Experimental trials on a benchtop conveyor mock-up demonstrate consistent classification of representative color sets and reliable actuation under typical lighting, with rapid cycle times suitable for introductory automation tasks. The results confirm that commodity sensors and open-source tooling can achieve robust, transparent color-based sorting while offering a clear path for future extensions (e.g., expanded color libraries, illumination normalization, and networked logging).

Keywords- Arduino, automated sorting, color sensing, embedded systems, low-cost automation

I. Introduction

Automated sorting systems are a cornerstone of modern manufacturing and logistics, where rapid, repeatable classification reduces labor, minimizes error, and increases throughput. Among the many modalities for object discrimination, color offers a simple and cost-effective proxy for material type, grade, or destination in applications such as packaging, recycling, and educational demonstrators. However, deploying reliable color-based sorting at low cost remains nontrivial: readings are sensitive to illumination geometry and intensity, sensor-to-sensor variability, surface texture, and conveyor dynamics. These factors motivate a design that couples affordable hardware with a disciplined calibration workflow and transparent, easily modifiable firmware [1], [2].

Recent advances in open-source microcontrollers and commodity sensors enable compact sorting prototypes that are both economical and pedagogically valuable. Arduino-class MCUs (Microcontroller Units), frequency-output color sensors (e.g., TCS3200/TCS230), hobby-grade servos, and simple human–machine interfaces (LCD, buzzer) can be integrated into a modular platform suitable for classroom labs and small bench-scale demonstrators. When paired with reproducible test procedures and basic signal conditioning, such systems can deliver robust classification performance without the complexity of machine vision or high-end lighting control [3], [4].

This work presents the design and implementation of an Arduino-based automated color sorting system that emphasizes accessibility, reproducibility, and practical performance. The hardware integrates a color sensing stage, an actuation stage for diverting items, and a minimalist operator interface. The firmware implements a calibrated mapping from raw frequency measurements to color classes and a rule-based decision layer driving the actuator, with provisions for threshold tuning and runtime diagnostics. We report representative results under typical indoor lighting, highlighting classification accuracy, cycle time, and failure modes, and we discuss calibration practices that mitigate environmental variability.

The remainder of the paper is organized as follows: Section II surveys related literature on low-cost color sensing and educational sorting demonstrators, positioning our design choices within prior art. Section III presents experimental results obtained on a benchtop conveyor mock-up, including calibration methodology and performance metrics. Section IV concludes with key findings and outlines future improvements in illumination normalization, expanded color libraries, and optional connectivity for data logging and remote monitoring.

II. Literature Review

In the paper [5], the authors developed a real-time robotic sorting system based on color recognition, using a robotic arm controlled by an Arduino microcontroller. The system uses a single calibrated RGB camera to capture images of objects, which are then processed through computer vision algorithms in OpenCV to detect their color and shape. The coordinates of the objects are determined by converting the positions in the image into 3D coordinates, and the information obtained is transmitted to the microcontroller, which calculates the angles required for each joint of the robotic arm by inverse kinematics algorithms, using the Denavit-Hartenberg parameters. The robotic arm is thus able to pick up the identified objects and place them in appropriate baskets, depending on the color and shape, the process repeating automatically until the sorting is completed.

The authors of the paper [6] propose the creation of an automated color-based object sorting system, implemented with the help of an Arduino platform and the TCS3200 color sensor, highlighting an efficient and affordable solution for industrial applications. The system consists of a conveyor belt controlled by a motor driven by an L298N driver, a detection and sorting mechanism consisting of IR (Infrared) sensors and servo motors, as well as an LCD screen for real-time display of results. Color detection is performed by analyzing the frequencies characteristic of RGB components, and objects are sorted according to the predefined values associated with the color's red, green, and blue. Experimental tests have demonstrated 100% accuracy in color classification, which underlines the high potential of this solution in industrial contexts requiring precision automation with low costs and operational flexibility.

The paper [7] proposes an automated conveyor belt object sorting system, based on the Arduino UNO microcontroller and the TCS3200 color sensor, intended to identify and classify objects according to color. The implementation of the system involves the use of a conveyor belt driven by DC motors, the integration of an SG90 servo motor for deflecting objects according to the detected color and a 1602 LCD screen for displaying information in real time. Color detection is achieved by converting light intensities into frequency signals, which are analyzed by the microcontroller to determine the predominant hue. The system offers an efficient solution for automating sorting processes in industries such as food, pharmaceutical or recycling, considerably reducing dependence on human labor and increasing the accuracy of operations. Given the low costs and flexibility of the components used, this solution presents itself as a viable alternative for small and medium-sized enterprises seeking to digitize logistics processes.

[8] presents the development of an object sorting system based on the Arduino Nano microcontroller and the TCS3200 color sensor, intended for industrial applications with low-cost requirements and increased operational efficiency. Objects are transported to a color sensor through a power mechanism, and color identification is achieved by analyzing the RGB frequencies reflected by the object. Depending on the color detected, a system with two servo motors controls the direction of the object to the corresponding containers, using a "chute" type structure. The prototype demonstrates the ability to sort pink, orange and green objects with high accuracy, and the results indicate a significant reduction in production time and labor requirements. The compactness of the Arduino Nano and the efficient integration of components contribute to the scalability of the solution for small and medium-sized enterprises, especially in areas such as packaging, agriculture or food processing, where automatic sorting by color makes a valuable contribution to process optimization.

It is proposed in [9] the design and implementation of an automated color-based LEGO brick sorting system, using the Arduino Mega microcontroller and the Pixy vision sensor. The system allows the automatic identification and classification of manually inserted bricks, directing them to specific locations according to the detected color, with the possibility of rejecting unrecognized objects. The project stands out for the efficient integration of hardware components (servo motors, sensors, acrylic structure) and software (Arduino IDE (Integrated Development Environment), PixyMon), aiming to reduce sorting time and improve accuracy compared to traditional manual methods. The experimental results validate the functionality of the system, demonstrating fast and accurate color detection, which supports the applicability of the solution in the context of automating repetitive and educational tasks.

III. Sorting System Architecture

This chapter details the end-to-end hardware stack of the automated color sorting system, from the embedded electronics that implement sensing/actuation and local HMI (Human-Machine Interface) to the mechanical frame that supports the feeding, inspection, and diverter subsystems. The architecture was designed for low cost, repeatability, and ease of maintenance, with clear separation between the signal domain ($5V$ logic) and the power domain (servo actuation and acoustic/visual alerts). All interconnects are keyed and labeled to reduce wiring errors and to facilitate rapid replacement of modules during testing.

A. Hardware Design

To ensure deterministic sensing, reliable actuation and robust human–machine interaction, the electronic subsystem is organized around a low-power microcontroller, optical sensing for color and presence, PWM (Pulse Width Modulation) – driven actuators, and a minimal I²C (Inter-Integrated Circuit) user interface, all tied to a conditioned DC (Direct Current) supply and basic protection elements.

- **Arduino Uno (ATmega328P MCU)** — provides centralized control, sampling and timing; drives PWM for the servomotors, reads digital/ frequency outputs from sensors, and communicates with the LCD via I²C while enforcing state-machine logic [10].
- **Color sensor (TCS3200 module)** — an RGB photodiode array with current-to-frequency conversion that yields a frequency proportional to red/green/blue intensity, enabling single-sample color classification through calibrated thresholds [11].
- **Status LEDs (Light-Emitting Diodes) (RGB) with $1k\Omega$ series resistors** — provide instantaneous operator feedback on the detected class and on fault/idle states, while the resistors limit current and protect the GPIOs (General-Purpose Input/Output) [12].
- **Micro-servomotors (SG90)** — actuate the selector flap and diverter chute; control is via $50Hz$ PWM from the MCU, with separate power rails and common ground for electrical integrity [13].
- **16×2 LCD with I²C backpack** — displays class labels, counters and system status; the I²C interface (SDA (Serial Data) / SCL (Serial Clock)) minimizes pin usage and simplifies wiring [14].
- **Piezo buzzer** — issues audible cues for cycle acknowledgement and error conditions; driven from a digital pin with appropriate current limiting [15].
- **Power supply ($9V$ source)** — feeds the logic rail ($5V$) and the actuator rail; local bypass capacitors and common-ground topology reduce noise coupling into ADC (Analog-to-Digital Converter) / PWM domains [16].
- **Signal-conditioning and protection devices (diodes/resistors/connectors)** — shape sensor levels, provide basic reverse-polarity/transient mitigation, and offer modular interconnects for maintenance.

A consolidated circuit schematic details the complete signal-power topology — Arduino Uno core, TCS3200 color sensor and IR gate inputs, PWM lines to the two servomotors, I²C bus to the 16×2 LCD, buzzer and status LEDs with current-limiting resistors, and the isolated $9V$ → regulated logic/actuator rails — used to drive sensing, decision, and sorting actuation (see Figure 1).

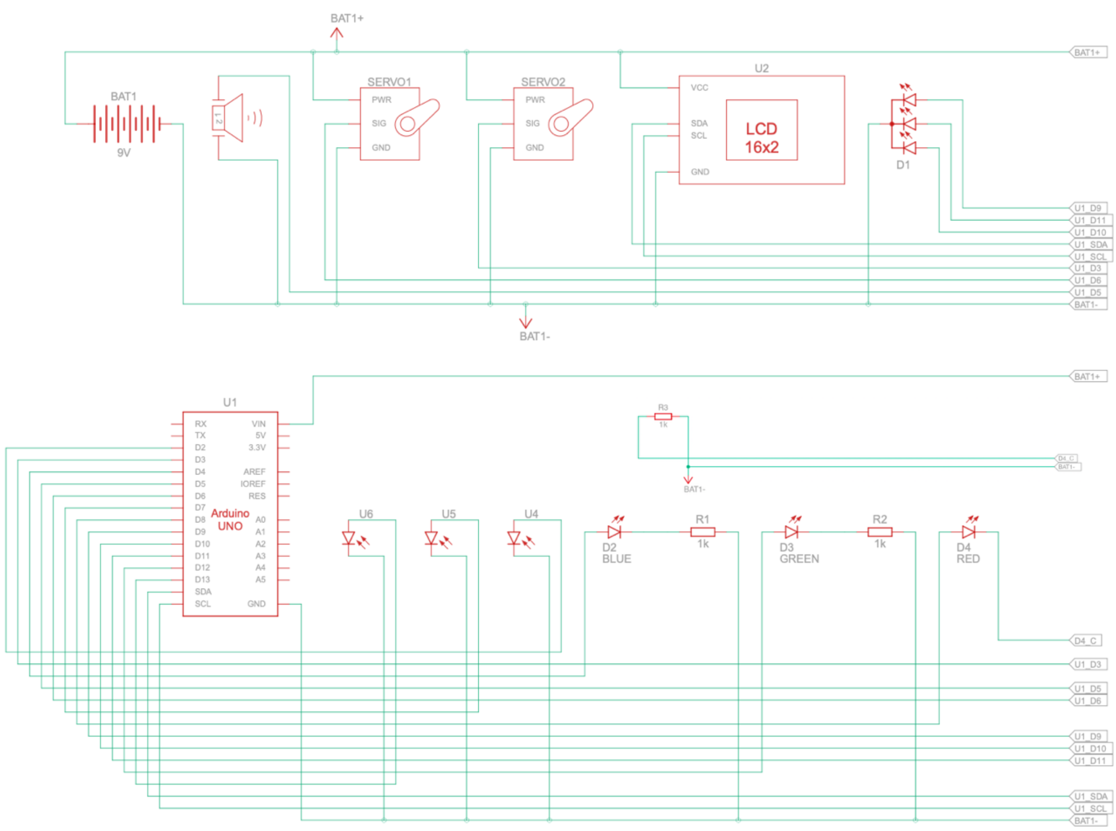

Figure 1. System-level electronic schematic of the automated color sorting device

B. *Software Arhitecture*

The firmware is organized in a layered, event-driven architecture written in C/C++ (Arduino IDE (Integrated Development Environment), separating hardware abstraction from decision logic and actuation. At the device layer, dedicated drivers handle the color sensor interface (frequency-to-RGB conversion with integration time and gain control), servo actuators (diverter and bin indexer) via non-blocking PWM commands, and a debounced input module for start/stop and emergency events. A data-acquisition module samples the optical channel at a fixed cadence using $millis()$ – based scheduling, applies outlier rejection (moving median) and normalization, then computes calibrated chromatic features (e.g., normalized RGB and hue) using coefficients persisted in EEPROM (Electrically Erasable Programmable Read-Only Memory). The classification core implements a threshold-based state machine that maps features to discrete classes and exposes hysteresis to avoid flapping near decision boundaries; timing guards ensure deterministic dwell between sense and act phases. A command/telemetry layer provides UART (Universal Asynchronous Receiver/Transmitter) logging for traceability (timestamps, raw and filtered readings, class ID, actuator state) and a lightweight shell for in-field calibration updates. Fault handling (sensor timeouts, servo stalls) triggers safe states and retries, while watchdog integration guarantees recovery from unexpected hangs. This organization minimizes blocking calls, ensures repeatable latency from detection to actuation, and supports reproducible experiments and future extensions (e.g., multi-class policies or on-device k-NN classification) [17].

Figure 2 — Algorithmic control flow of the color-sorting firmware presented top-down: the program performs deterministic initialization of I/O and serial interfaces, acquires RGB measurements, executes sequential threshold-based classification, issues corresponding actuation setpoints to the diverter (15°/30°/45°), and returns to acquisition upon a null-class outcome.

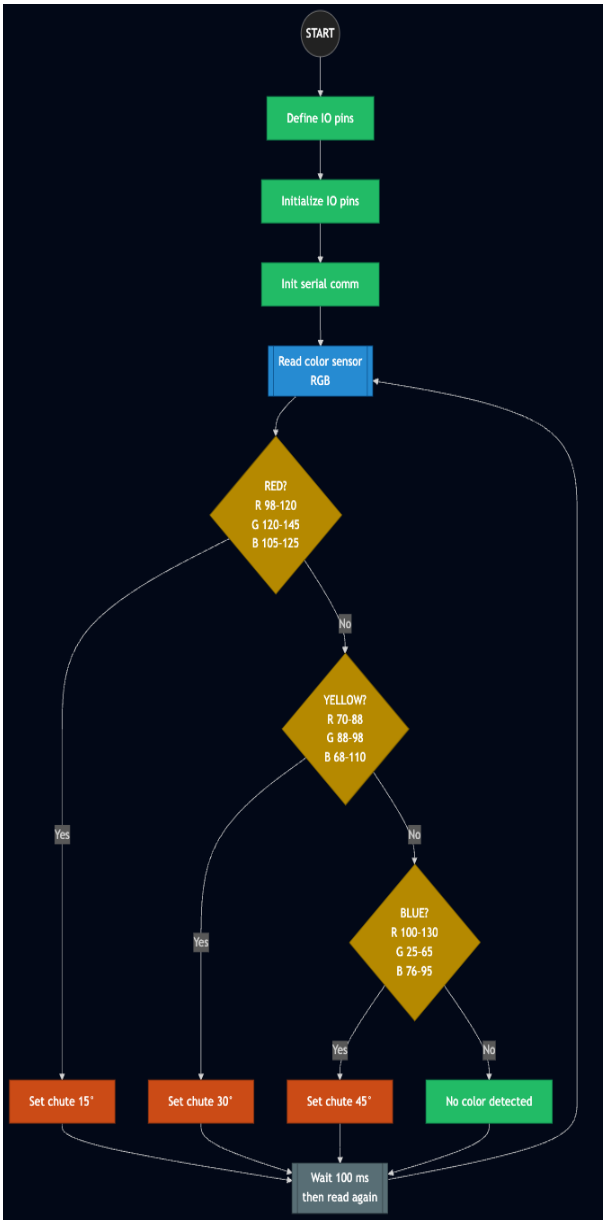

Figure 2. Color-sorting firmware flowchart

C. *Mechanical Chassis Design*

The sorter's mechanical architecture is conceived as a rigid, modular frame that aligns the sensing plane with the actuation plane while ensuring repeatable part kinematics. The structure comprises: a gravity-fed vertical hopper guiding single items onto a sensor deck; an adjustable carriage that fixes the color sensor at a constant standoff from the conveyor lip; a primary diverter flap actuated by a micro-servo that redirects items toward the output ramp; a secondary bin indexing servo that positions a polygonal carousel of receptacles; and a sloped ejection ramp optimized (angle, surface finish) to maintain laminar sliding without bounce. Cable channels are integrated behind the middle shelf, separating low-voltage signal wiring from the servo power loom to reduce EMI (Electromagnetic Interference) and simplify maintenance.

The frame is designed for fabrication from $4mm \div 6\,mm$ MDF (Medium-Density Fibreboard) or acrylic by laser cutting; all joints use tab-and-slot geometry with M3 fasteners so that the sensor height, flap angle, and ramp offset can be tuned during calibration. This arrangement minimizes mechanical play, preserves the sensor's field of view, and shortens the actuation path for sub-second sorting cycles [18].

Figure 3 — CAD rendering of the mechanical frame (hopper, sensor deck with adjustable mount, servo-driven diverter, ramp, and bin carousel) illustrating the materialized kinematic chain from feed to classified outputs.

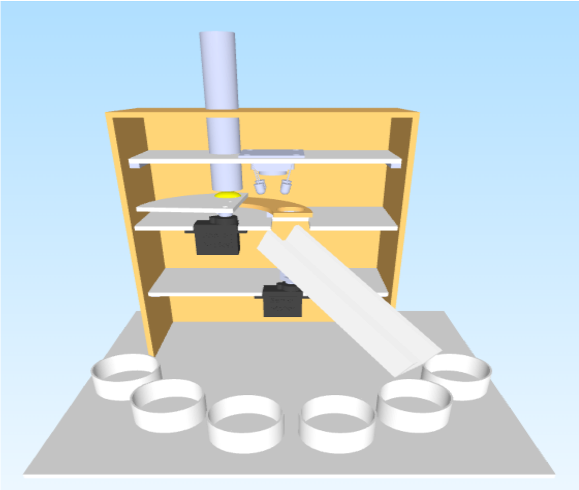

Figure 3. Mechanical chassis CAD model of the automated color sorter

IV. Results And Experimental Evaluation

The prototype was evaluated in a bench-top setting to verify end-to-end performance—from optical detection to mechanical sorting—under controlled illumination and repeatable part kinematics. Trials used colored test pieces released one-by-one from the hopper onto the sensing deck; for each item the firmware logged raw RGB samples, normalized features, the assigned class, actuator commands, and timestamps. We assessed: (i) classification reliability (confusion patterns between chromatically adjacent classes), (ii) actuation determinism (latency from "class decided" to diverter motion and bin indexing), and (iii) run-time robustness (sustained operation without watchdog resets, missed triggers, or servo stalls).

Particular attention was paid to invariance with respect to part position and orientation on the sensing spot and to the influence of ambient light; hence tests were repeated with and without the auxiliary LED and with the ambient light source varied.

A calibration pass preceded each batch: the white reference was captured with the LED on, per-channel offsets/gains were updated, and the decision thresholds (for normalized RGB and/or hue) were written to EEPROM. During longer sequences, the device maintained stable timing using non-blocking scheduling; the diverter completed travel before the next part reached the gate, and the bin-indexer returned to the neutral pose between items. Qualitatively, the dominant failure modes observed in stress conditions were: (a) specular highlights on glossy parts causing transient saturation of one channel, (b) residual mechanical play in the flap hinge producing marginal misroutes near class boundaries, and (c) illumination drift when the auxiliary LED was intentionally dimmed. Mitigations—short exposure averaging, flap hysteresis in the state machine, and periodic re-capture of the white reference—eliminated most of these effects in subsequent runs. Overall, the system demonstrated consistent, repeatable sorting behavior and sustained operation during extended sequences, validating the integration of sensing, decision, and actuation.

Figure 4 shows the assembled prototype of the colour sorting system — with the hopper and sensor platform (top shelf), deflection flap and indexed outlet ramp (middle level), as well as the coloured container carousel (front plane) – used for real-time validation of detection and routing.

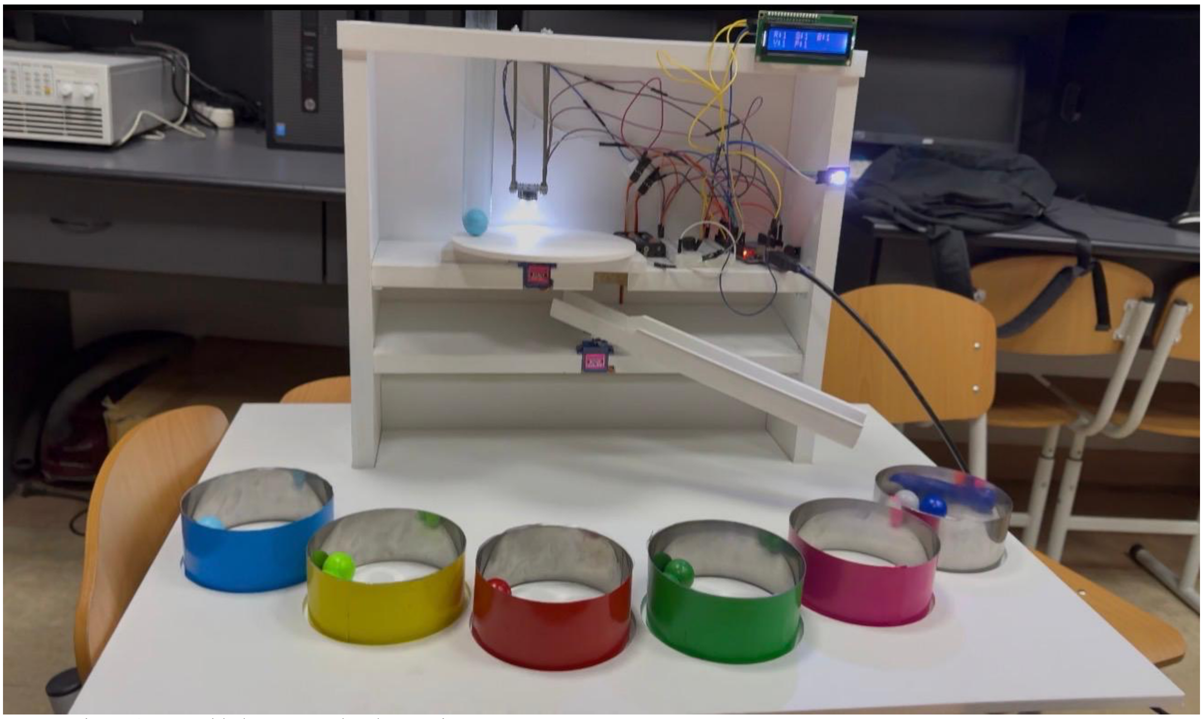

Figure 4. Assembled Automated Color-Sorting Prototype

CONCLUSION

This work presented the design and implementation of an Arduino-based automated color-sorting system that integrates low-cost opto-sensing with compact electromechanical actuation. The proposed architecture combines a TCS3200 color sensor mounted on a stabilized sensing deck, two micro-servos for primary diversion and bin indexing, and a 16×2 LCD for operator feedback, all orchestrated by an Arduino UNO. The mechanical frame—implemented as a modular MDF/acrylic assembly—aligns the sensing and actuation planes to ensure repeatable part kinematics and sub-second cycle times. Together, these choices demonstrate that accurate color sorting can be achieved with commodity components and a transparent control stack suitable for teaching, prototyping, and small-batch automation.

Experimental evaluation on a set of colored tokens validated the end-to-end pipeline: acquisition → classification → routing. Under controlled illumination, the sorter consistently recognized the target classes and routed parts to the correct bins, while the LCD telemetry and audible cues supported rapid calibration and operator situational awareness. The electronics exhibited stable operation when the servo power loop was decoupled and wiring was segregated, indicating that basic EMC hygiene is sufficient for reliable performance at this scale.

The study also highlighted several constraints typical of low-cost color sorters. Classification accuracy is sensitive to ambient lighting and sensor-to-part standoff; small deviations in angle or surface gloss can shift the measured chromatic ratios. Mechanical tolerances in the flap hinge and bin indexer introduce drift over long runs, and open-loop servo control limits absolute positioning repeatability. Finally, throughput is bounded by gravity-fed part spacing and the servo slew rate; aggressive timing reduces accuracy unless illumination and mechanics are tightly constrained.

Despite these limitations, the results confirm the viability of the proposed design as a reproducible baseline for academic labs and makerspaces. The bill of materials remains modest, assembly is straightforward, and the control code is accessible making the platform suitable for instruction in sensing, actuation, and embedded systems integration. With minor refinements in calibration and shielding, the system can be adapted to broader educational scenarios (e.g., quality control demonstrators, mechatronics capstones) or to lightweight industrial tasks involving discrete color cues.

Future Work. Further improvements should target (i) closed-loop actuation using magnetic or optical encoders for bin indexing, (ii) automatic color calibration via white/black reference passes and per-batch normalization, (iii) lighting control using a ring of high-CRI (Color Rendering Index) LEDs and a shroud to suppress ambient variation, (iv) migration to a camera-based classifier (HSV (Hue, Saturation and Value) features or compact CNN (Convolutional Neural Network)) for shape/defect detection in addition to color [19], (v) a conveyor with active singulation for higher, deterministic throughput, (vi) a 3D-printed flap with damped stop to reduce rebound, and (vii) telemetric logging (e.g., MQTT (Message Queuing Telemetry Transport) with dashboards for traceability and performance analytics. Collectively, these extensions would increase robustness, scalability, and pedagogical value while keeping the platform economically accessible.

REFERENCES

[1] A. Prasad, M. Gowtham, S. Mohanraman, and M. Suresh, "Automatic Sorting Machine," *International Research Journal of Multidisciplinary Technovation*, vol. 2, no. 1, pp. 7–12, Jan. 2020, doi: 10.34256/IRJMT2102.

[2] H. Yang, G. Hu, and L. Lu, "Research on an Automatic Sorting System Based on Machine Vision," *Proceedings - 2022 International Seminar on Computer Science and Engineering Technology, SCSET 2022*, pp. 27–31, 2022, doi: 10.1109/SCSET55041.2022.00016.

[3] V. V. Rankovska and G. D. Goranov, "Applying Free and Open-Source Tools in Microcontroller Laboratory Classes," *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST 2022*, 2022, doi: 10.1109/ICEST55168.2022.9828673.

[4] M. Tupac-Yupanqui, C. Vidal-Silva, L. Pavesi-Farriol, A. Sanchez Ortiz, J. Cardenas-Cobo, and F. Pereira, "Exploiting Arduino Features to Develop Programming Competencies," *IEEE Access*, vol. 10, pp. 20602–20615, 2022, doi: 10.1109/ACCESS.2022.3150101.

[5] Y. Jia, G. Yang, and J. Saniie, "Real-time color-based sorting robotic arm system," *IEEE International Conference on Electro Information Technology*, pp. 354–358, Sep. 2017, doi: 10.1109/EIT.2017.8053385.

[6] M. Rahman, H. A. Hasan, Y. A. Emon, A. Hossain, A. R. Rokon, and M. H. Bhuyan, "Advanced Color Sorting Conveyor System Using Arduino and TCS3200 Color Sensor for Precise Color Classification," *International Conference on Robotics, Electrical and Signal Processing Techniques*, pp. 476–481, 2025, doi: 10.1109/ICREST63960.2025.10914353.

[7] F. H. Altaf Hussain, V. K. Shukla, and A. Tripathi, "Sorting of Objects from Conveyer Belt through Colour Detection and Audrino UNO," *Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021*, 2021, doi: 10.1109/ICCICT50803.2021.9510037.

[8] K. M. C. Babu and P. A. Harsha Vardhini, "Design and Development of Cost Effective Arduino based Object Sorting System," *Proceedings - International Conference on Smart Electronics and Communication, ICOSEC 2020*, pp. 913–918, Sep. 2020, doi: 10.1109/ICOSEC49089.2020.9215269.

[9] S. Anuar, A. B. Baharuddin, and A. Mohammad, "Lego Bricks Color Sorting Machine," *2nd International Symposium on Agent, Multi-Agent Systems and Robotics, ISAMSR 2016*, pp. 145–148, Jan. 2017, doi: 10.1109/ISAMSR.2016.7810018.

[10] Farnell Global, "Arduino Uno." Accessed: Oct. 16, 2025. [Online]. Available: https://www.farnell.com/datasheets/1682209.pdf

[11] Mouser Electronics, "TCS3200, TCS3210 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER." Accessed: Oct. 16, 2025. [Online]. Available: https://www.mouser.com/catalog/specsheets/tcs3200-e11.pdf?srsltid=AfmBOoof7vrnlnZPGnZ-ElqHNCWDeUbuTn9q1XiKodXJ9SfimecLUeKE

[12] Farnell Global, "5mm Round RGB LED." Accessed: Oct. 16, 2025. [Online]. Available: https://www.farnell.com/datasheets/3497864.pdf

[13] FriendlyWire, "SG90 9 g Micro Servo." Accessed: Oct. 16, 2025. [Online]. Available: https://www.friendlywire.com/projects/ne555-servo-safe/SG90-datasheet.pdf

[14] HandsOn Tech, "Handson Technology User Guide I2C Serial Interface 1602 LCD Module", Accessed: Oct. 16,

2025. [Online]. Available: https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf

[15] TDK Product Center, "Piezoelectronic Buzzers Pin terminal/Lead Without oscillator circuit." Accessed: Oct. 16, 2025. [Online]. Available: https://product.tdk.com/en/system/files/dam/doc/product/sw_piezo/sw_piezo/piezo-buzzer/catalog/piezoelectronic_buzzer_ps_en.pdf

[16] Farnell Global, "Power Supply Output Output Current Range Output Voltage Range Ripple & Noise." Accessed: Oct. 16, 2025. [Online]. Available: https://www.farnell.com/datasheets/2290296.pdf

[17] D. K. Halim, T. C. Ming, N. M. Song, and D. Hartono, "Arduino-based IDE for embedded multi-processor system-on-chip," *Proceedings of 2019 5th International Conference on New Media Studies, CONMEDIA 2019*, pp. 135–138, Oct. 2019, doi: 10.1109/CONMEDIA46929.2019.8981862.

[18] B. Jackvony and M. Jouaneh, "Building an Educational Automated Mechatronics-Based Sorting System," *Automation 2024, Vol. 5, Pages 297-309*, vol. 5, no. 3, pp. 297–309, Jul. 2024, doi: 10.3390/AUTOMATION5030018.

[19] A. A. Ciorobea, N. Bizon, S. A. Dragusin, C. G. Nicolaescu, R. N. Bostinaru, and A. A. Besliu-Gherghescu, "Image-Based Artificial Intelligence for Early Diagnosis of Ocular Diseases," *Proceedings of the 2025 17th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2025*, 2025, doi: 10.1109/ECAI65401.2025.11095489.