

Sweet Sort

Introduction

Student: Iliescu Laura-Gabriela Group 1221A

Sweet Sort is an exciting Arduino project that allows you to build your own color candy sorting machine. With this machine, sorting large quantities of candies by color becomes a quick and easy process. This project is an excellent way to learn about Arduino programming and electronics while creating a fun and useful device.

As an Arduino enthusiast, I find this project to be both challenging and rewarding. Not only do you get to create a useful device, but you also gain a deeper understanding of Arduino programming and electronics. Building Sweet Sort is a great way to expand your skillset and showcase your creativity.

General Description

With just one TCS230 color sensor and two SG90 servomotors, you can easily sort candies based on their color. The machine operates by moving the candy from the main container to the sensor and then to its designated container based on its color. To make the sorting process even more seamless, the machine comes equipped with an LCD screen with I2C Interface that displays the color of each candy. Additionally, an auditory alert system in the form of a buzzer will notify you when a candy has been successfully sorted. The project also features RGB LEDs that illuminate to display the color of the candy.



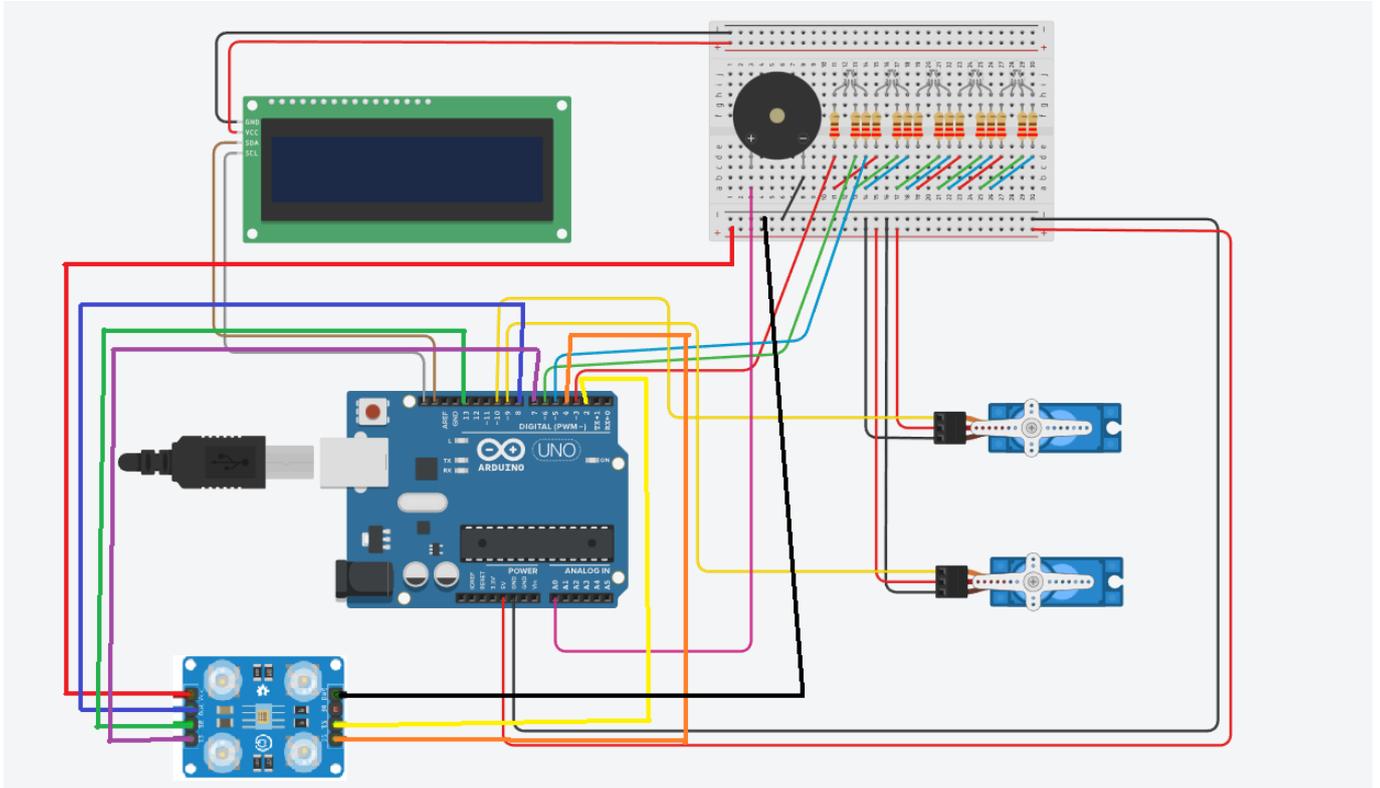
Hardware Design

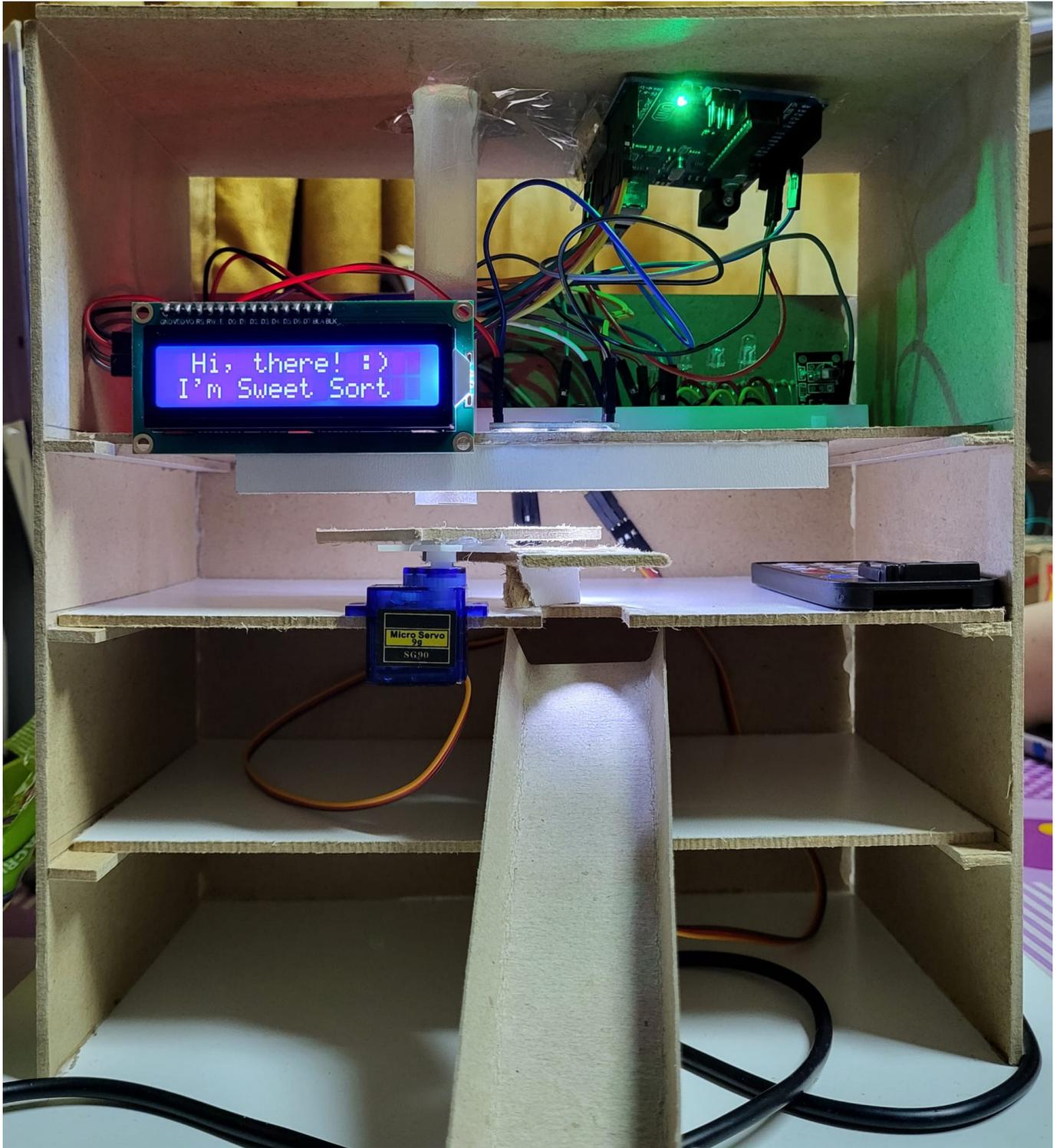
The list of components for this project includes:

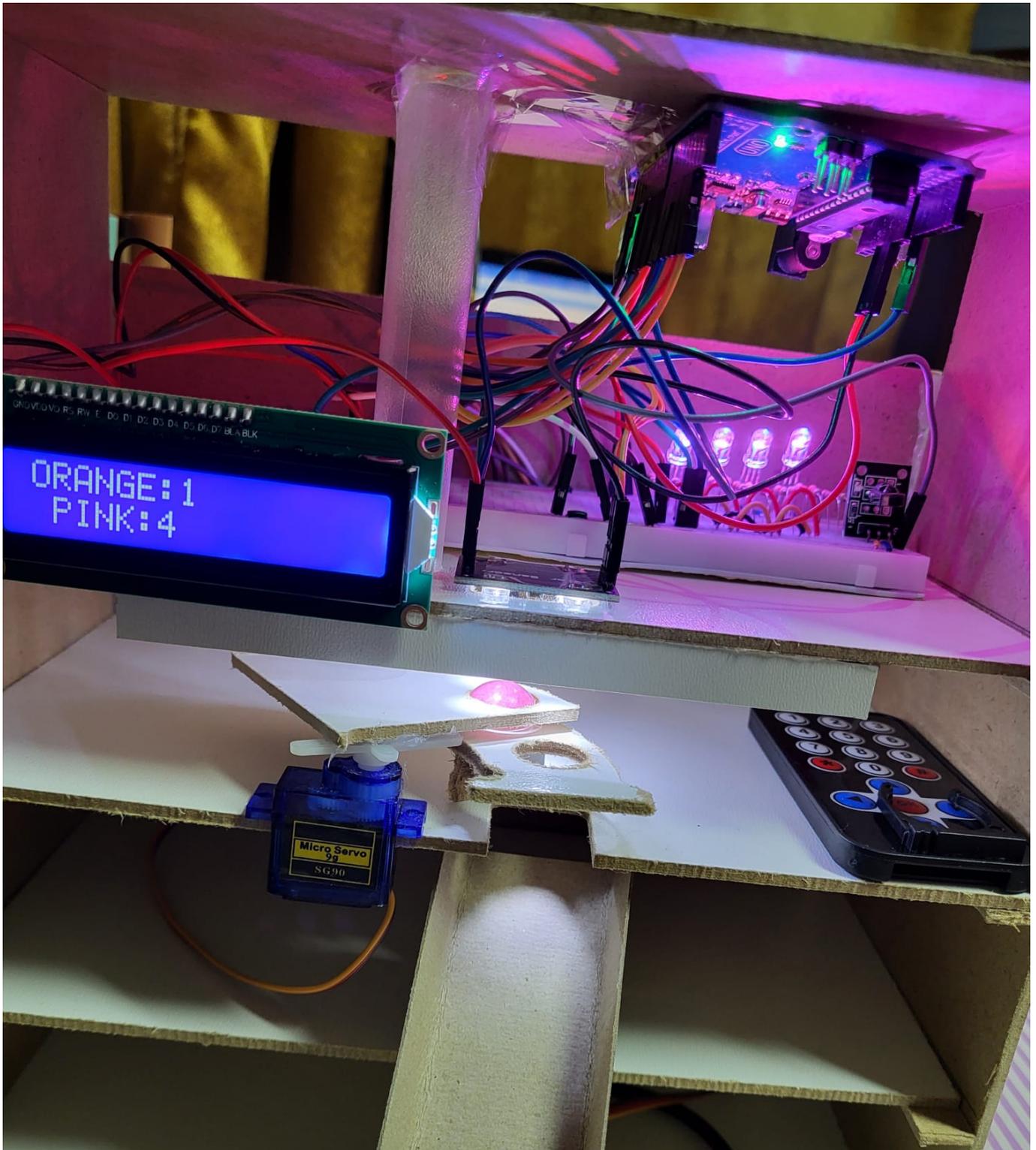
- Arduino UNO R3
- TCS230 color sensor
- 2 SG90 servo motors (180 degrees)
- LCD 1602 with I2C interface module

- Breadboard
- Buzzer
- Common cathode RGB LEDs
- Wires

For the box I used a 2.5mm thick MDF board







Software Design

Development environment: Arduino IDE

The code is an Arduino sketch that controls a candy sorting machine using a color sensor, servomotors, and an LCD display. Here's an explanation of the code:

1. Libraries: The code includes two libraries: Servo.h for controlling the servomotors and

LiquidCrystal_I2C.h for interfacing with the LCD display.

2. Pin Definitions: Various pins are defined using the `#define` directive for easy reference throughout the code. These pins are used to connect the color sensor, servomotors, RGB LED, buzzer, and LCD display.

3. Global Variables: The code defines several global variables to store calibration values, pulse width measurements, and the number of candies sorted for each color.

4. Setup Function: The `setup()` function is executed once at the beginning of the program. It initializes the serial communication, attaches the servomotors, sets pin modes, initializes the LCD, and sets the initial position for the top servomotor.

5. Loop Function: The `loop()` function is executed repeatedly after the `setup()` function. It controls the main operation of the candy sorting machine. The loop consists of the following steps:

a. Removing Candies: The top servomotor rotates from position 10 to 100, causing the candies to fall out of the tube. The delay of 15 ms after each position change allows the servomotor to reach the desired position.

b. Color Sensing: The top servomotor rotates from the maximum position (`TOP_MAX`) to the minimum position (`TOP_MIN`) while checking the color of the candy in front of the color sensor. When the servomotor reaches the position (`SENSOR_POS`) in front of the color sensor, it stabilizes for 1 second and reads the pulse widths of the red, green, and blue components using the color sensor.

c. Color Classification: The pulse width values are compared with the calibrated minimum and maximum values for each color (red, green, and blue). If a candy matches the color criteria, the corresponding RGB LED values, sound tone, and position for the bottom servomotors are set, and the count for that color is incremented.

d. LCD Display: The LCD is updated with the count of candies for each color.

e. RGB LED Control: The RGB LED is controlled by `analogWrite()` function, setting the appropriate values for the red, green, and blue pins based on the color of the candy being sorted.

6. Helper Functions: The code includes several helper functions:

a. `getRedPW()`, `getGreenPW()`, `getBluePW()`: These functions configure the color sensor to read the pulse width of a specific color (red, green, or blue) and return the measured pulse width value.

b. `playTone(unsigned int frequency, unsigned long duration)`: This function plays a tone on the connected buzzer for a specific frequency and duration.

The code continuously repeats the loop, allowing the candy sorting machine to sort candies based on their colors and display the count on the LCD.

Results

The candy sorting project using Arduino is a delightful endeavor that brings together various components to create a functional and interactive system. For an enthusiastic beginner venturing into the world of Arduino, this project serves as a rewarding experience, providing hands-on learning and an exciting avenue to explore the realm of electronics and automation. By integrating servomotors, a color sensor, an LCD display, and an RGB LED, beginners gain practical knowledge in wiring, programming, and module integration, honing their skills and nurturing their creativity.

Beyond the educational aspect, this project offers practical benefits. It introduces beginners to the concept of automation, showcasing how Arduino can be used to solve real-world problems. The candy sorting machine exemplifies the power of programming and sensor integration to accomplish a specific task, presenting beginners with the opportunity to delve into programming fundamentals and expand their problem-solving abilities. As they witness the RGB LED lighting up and the buzzer playing a tone when candies are sorted, the sense of achievement and satisfaction further fuels their enthusiasm.

[video_sweet_sort.10.zip](#)

Conclusions

In conclusion, the Sweet Sort project utilizing Arduino and a color sensor presents an engaging and technically-oriented endeavor for beginners. Through the integration of various components and programming techniques, this project offers practical insights into electronics, automation, and sensor calibration. The process of researching, acquiring components, designing the box, assembling the system, and working on the code contributes to a comprehensive understanding of hardware integration, precision engineering, and coding practices.

Throughout the project, the challenges of calibrating the TCS230 color sensor became evident due to its high sensitivity to changes in light conditions. This aspect provided valuable hands-on experience in troubleshooting, sensor optimization, and the importance of environmental factors on sensor accuracy. Overcoming these challenges required technical knowledge, perseverance, and experimentation to achieve the desired functionality and reliable color detection.

Download

[sweet-sort.zip](#)

Jurnal

25.04.2023 : I started researching for a project idea and when looking at different components available I saw a color sensor.

27.04.2023 : Decided on Sweet Sort as my project (since I was intrigued by the idea of working with a

color sensor).

28.05.2023 : I ordered the components for the project.

02.05.2023 : The components arrived and I started working on the box design, as it needed precise circular cuts for the candies to pass through.

16.05.2023 : Started assembling the components.

21.05.2023 : Worked on the code and attempted to calibrate the sensor. Discovered that anticipating the outcome of the TCS230 sensor was challenging due to its sensitivity to changes in light. 😞

28.05.2023 : Finally finished the project

Bibliografie/Resurse

LiquidCrystal_I2C.h:

https://ardushop.ro/ro/index.php?controller=attachment&id_attachment=57

Design inspiration:

https://www.youtube.com/watch?v=g3i51hdfLaw&ab_channel=HowToMechatronics

Color sensor calibration information:

https://www.youtube.com/watch?v=MwdANecTiPY&ab_channel=DroneBotWorkshop

<https://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/sweetsort>



Last update: **2023/05/30 16:04**