
# Overview

We have used the following components:

- Arduino Uno
- Color Sensor
- Computer running Processing

Our project consists of the following stages:

1) An Arduino Uno connected to a Color Sensor with a resistor in series.

2) A Processing powered script that plays a corresponding audio file between the 3 mapped colors.

The Arduino Uno is connected to a color sensor that detects the color in the RGB Color Space. The color sensor is actively powered and outputs infrared light. The Arduino Uno receives these values and then translates the RGB values into a single dominant color between Red Green and Blue. Accordingly, it transmits the dominant color over the serial port which is connected to the computer.

The computer is running a Processing script which is running on the same port as the Arduino which is connected via USB. Upon receiving the dominant color, it plays the corresponding audio file.

# Design Process

We initially designed the project in TinkerCAD. Following this, we mocked up the design using the Arduino Uno, some jumper cables, and a breadboard. Finally, we soldered the Arduino Uno after confirming that the circuitry was nominal. We have attached pictures of each stage.

First, we connect the Arduino Uno's  a0, a1, a2, and a3 pins to the p1, p2, p3, and p4 pins of the color sensor then connected the ground of the sensor to the ground of the Arduino  Uno followed by 5 volts to VCC but we noticed that the color sensor was getting headed up without any resistors so we had to connect it to the resistor first and then to the sensor

In the Arduino code, we are reading color sensor values in this Arduino code and using a speaker to play music based on the data. Adafruit_TCS34725.h for the colour sensor and SoftwareSerial for the speaker are two libraries we include to do this.  then define the pins for the speaker and the colour sensor. We utilize a constant for the speaker pin in addition to an array of sensor pins (A0 to A3). (9). We utilize the SoftwareSerial object to set up the speaker serial communication. Other pins outside the hardware serial port are available for communication with this device. (pins 0 and 1). We initialize the speaker serial communication as well as the serial communication for debugging in the setup method. As long as the Arduino board is turned on, the loop function is run. We read the color sensor readings and use a for loop and array to store them. Then, using data from the color sensor, We start playing music. We look at the readings from the several color sensors to see which color (red, green, or blue) has the greatest value. I use the Serial.print() method to print a relevant MP3 file name to the serial port based on the color (for debugging reasons).To prevent sudden changes in music playback and to allow for some time between readings, I add a 100-millisecond delay at the conclusion of the loop function. Overall, this code illustrates a straightforward use of an Arduino board and speaker to map colors to music using a color sensor but the close only outputs the names of the files these files' names are further connected to  code written in processing and then the audio file is played

The coding process was more involved, as it involved multiple rounds of troubleshooting and debugging. The primary cause for iterations was the process of fine-tuning the mapping for RBG values to determine the dominant color. A simple greater-than comparison did not suffice and we had to incorporate cut-off values to reduce the number of false positives

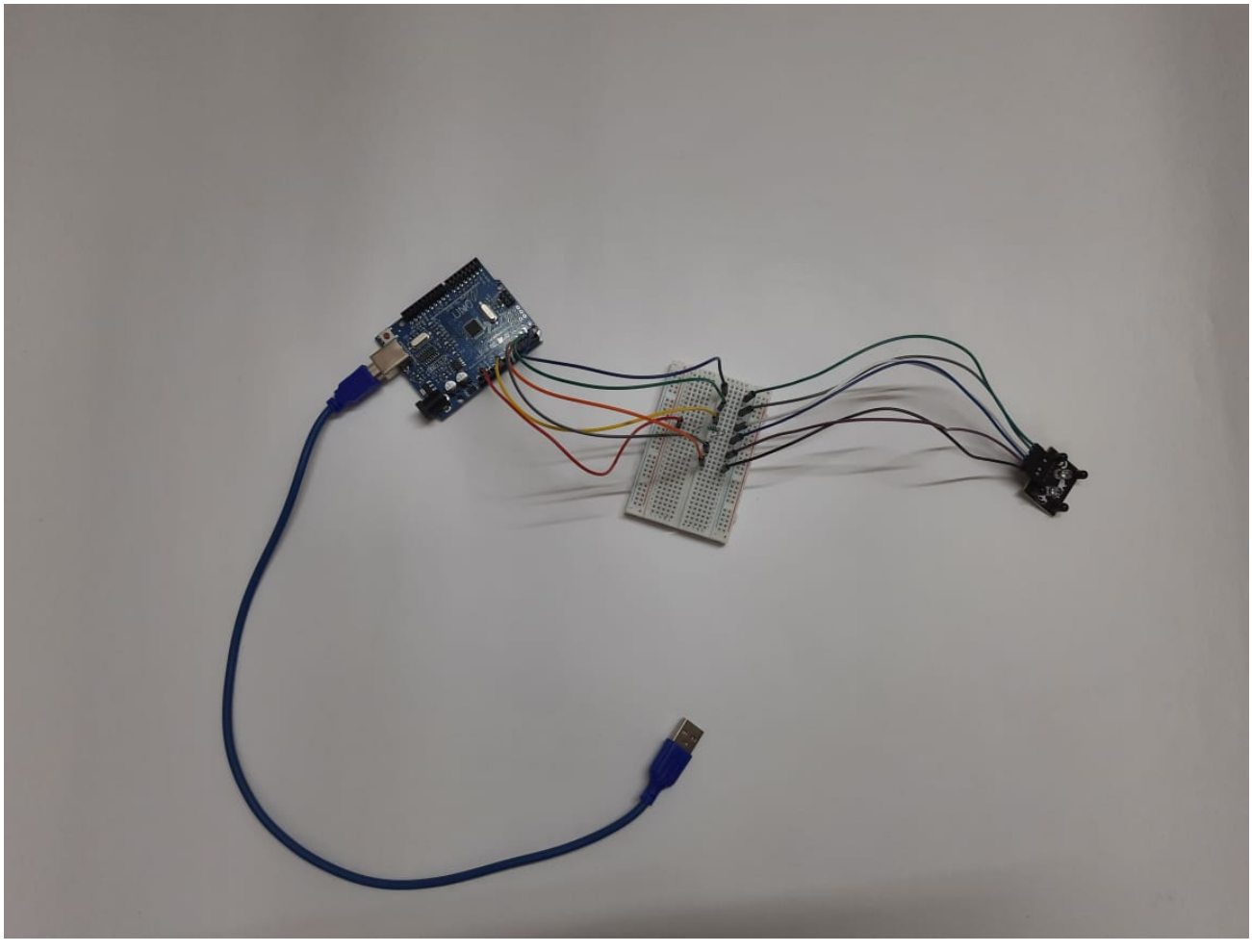
# Use Case

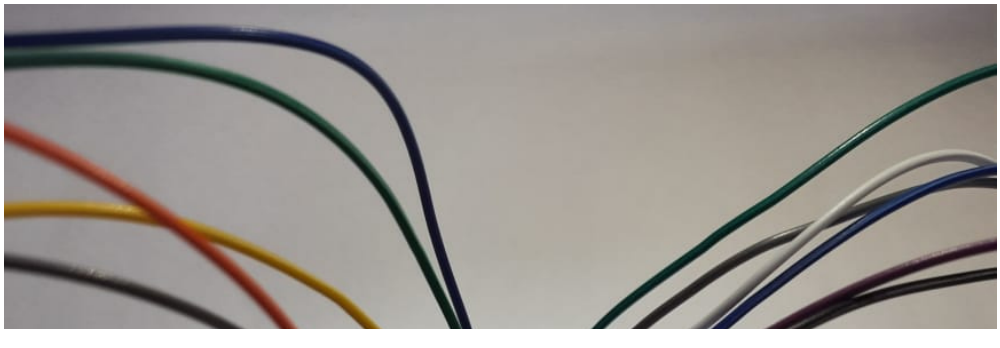
We conceptualized the following scenario which would ideally utilize our product:

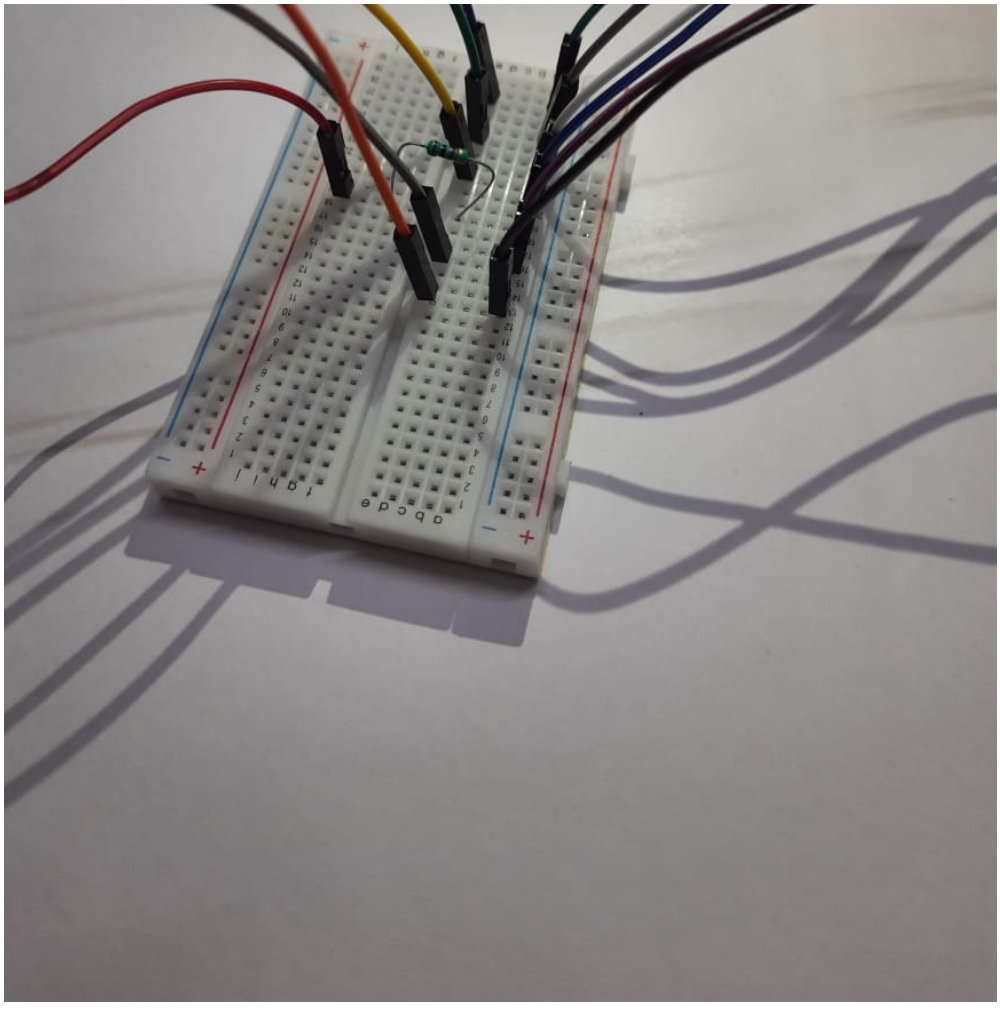
The end user installs the Arduino Uno in their office/reception area. The color sensor is pointed outwards. Between the range of color values of the sun, the product would play corresponding audio files throughout the day. For example during rainy weather, the dominant color would be blue: and hence slow paced elevator music would be played. Linking the music played in elevators and lobbies with the weather outside integrates the environment with the infrastructure and is not jarring to hear for the user.

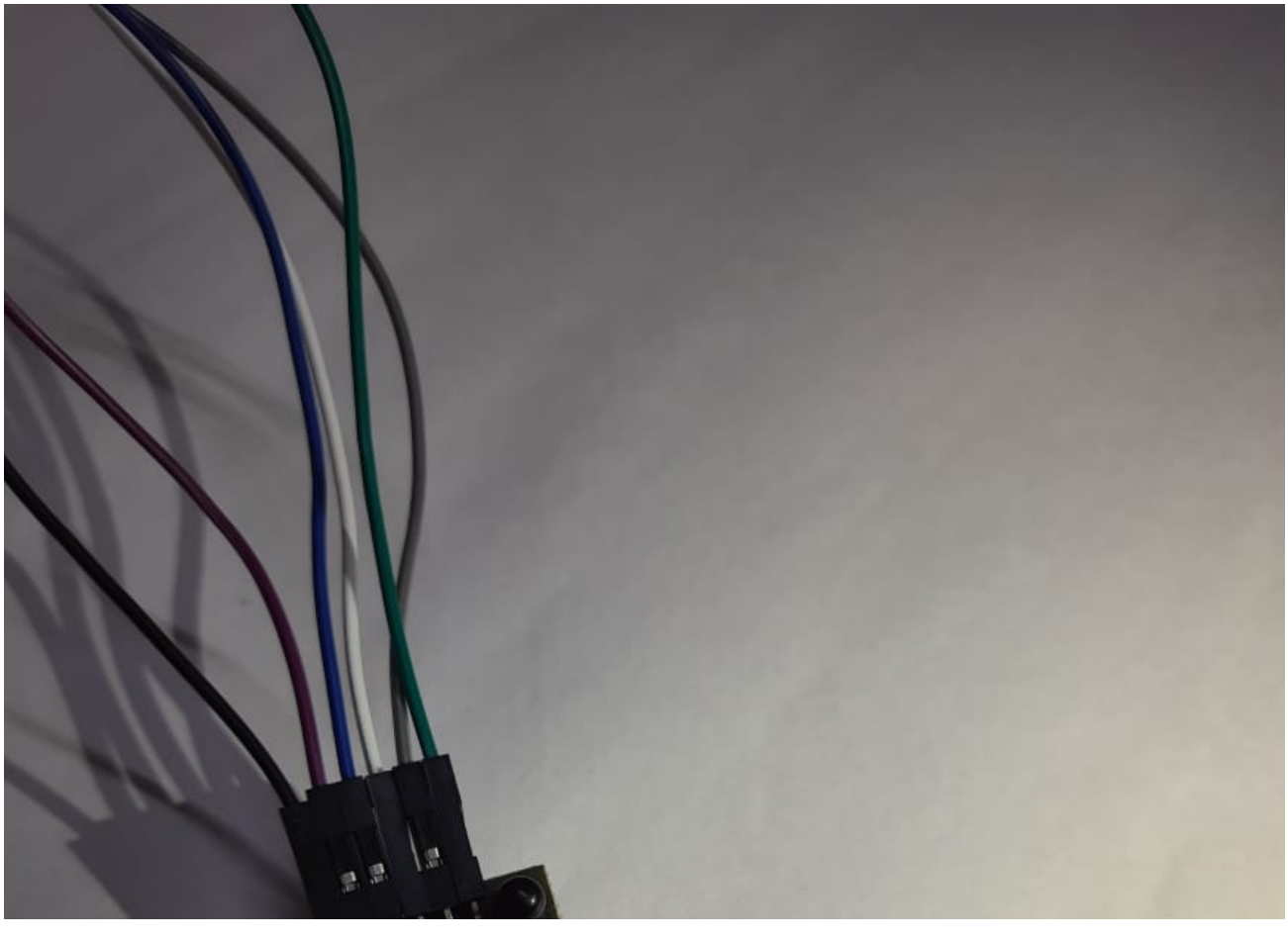
This product pushes users to be more aware of their surrounding environments. This additionally acts as an accessibility device for those who are blind since they would have a near subconscious understanding of the weather/time upon hearing the audio.

# Pictures

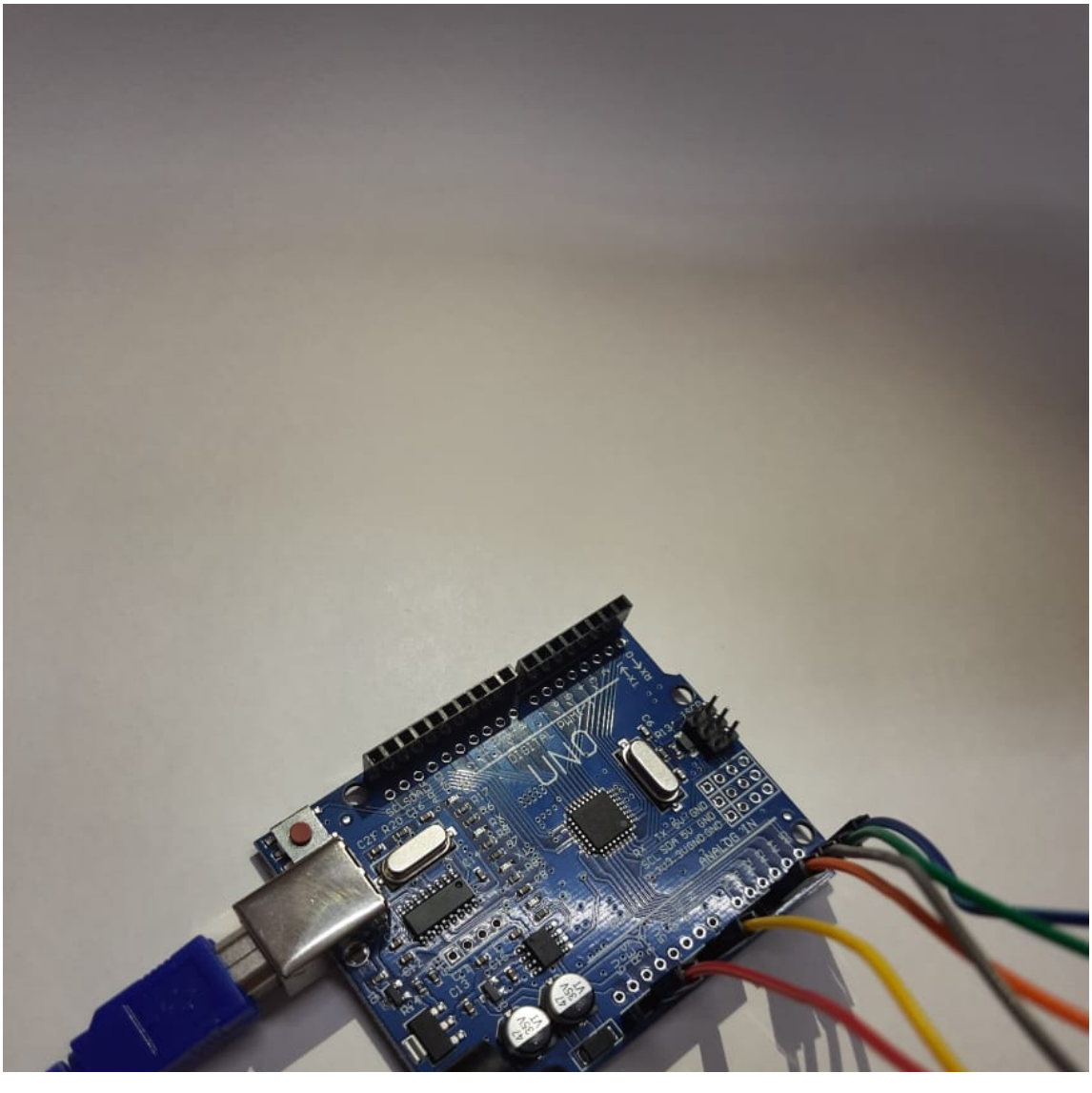


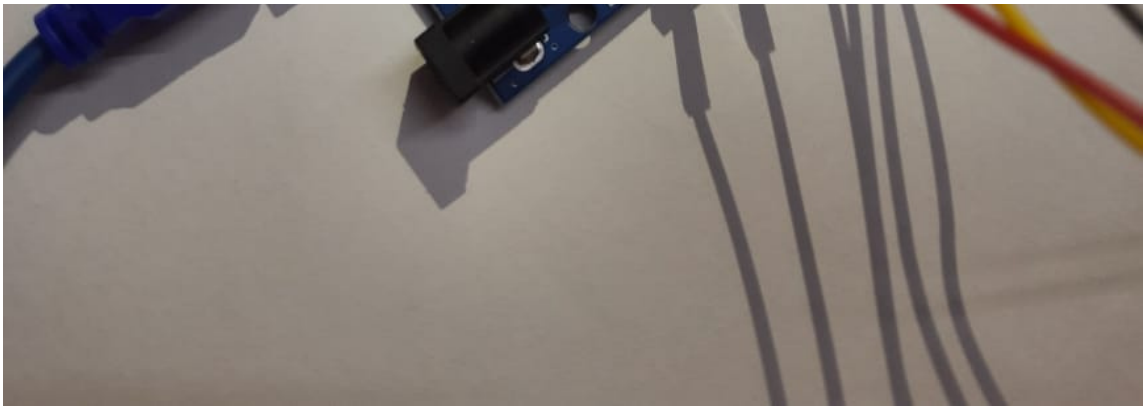


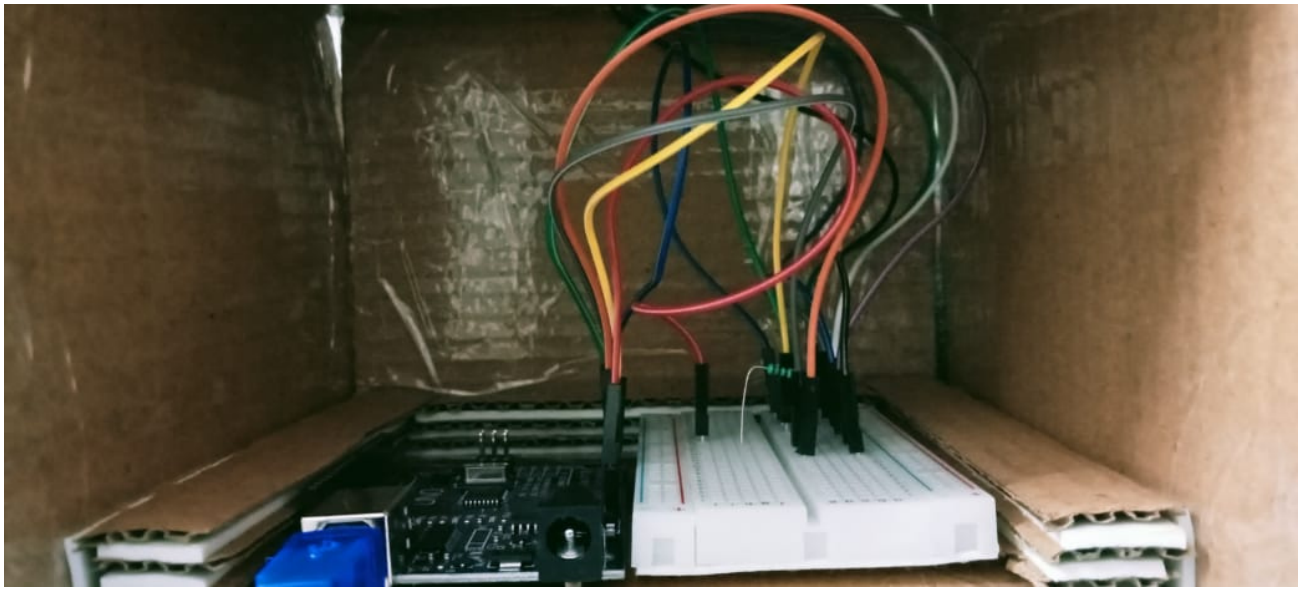





P1 P2 VCC GND P3


UNO






P1 P2 VCC GND P3 P