Color Sensing Stage

Hannah Holbert and August Menchaca
(E-Lab, Boston University)

(Dated: April 26, 2022)

We propose a color stage that will change color depending on the color of an object placed inside. The Arduino Uno will allow us to integrate an RBG color sensor, LED colored light strips, and a code that we create. A user will be able to place an object into a square foot stageso that the color sensor reads the wavelength of the objects color and maps it to RGB values. The RGB values are inverted and output the LED light strip that surrounds the color stage (back lit). .

I. INTRODUCTION

Many times when taking a picture of an object people have difficulty with staging it so it appears vibrant as opposed to the background. Our project will attempt to fix this issue by automatically by detecting the color of an object and making its background another color.

The design of this project will be a square stage that an object can be placed on. The RGB (Red Green Blue) color sensor is below the stage and will read the wavelength of the object on the stage through opaque plastic. The Arduino and is situated below the photo stage and is connected to both a computer with code and to the LED light strip. The LED light strip wraps around the walls of the stage and are fixed to a cardboard box that provides structure to the design. The LED light strip illuminated the stage through the opaque plastic.

II. PARTS

- Color sensor (TCS230 TCS3200)
- Opaque glass or plastic
- LED Strip
- Arduino Uno
- Cardboard frame

III. CODING

The final product we build required three separate portions of code to satisfy all of our demands. The first group of code was a simple test code with a function that only output assigned RGB values to the LED light strip. This was useful for testing the light strip and the perceived color of some RGB combinations. The next group of code worked by reading inputs of wavelength from the color sensor. This code read wavelengths and we would test and make not of the wavelengths for strong red, green, and blue observation. For this we ran the code and recorded values when red was presented directly to the sensor and then recorded values for which red was not at all present. These values became the threshold values that would be used to map wavelengths to RGB values.

The final code that is used is a ode that nearly combines the two previous codes. The color sensor inputs wavelengths that are mapped to the RBG values for each color, ranging from 0 to 225 (0 correlates with none of that color, 255 is that color LED turned all of the way on). If the RGB values are greater than 255, the code finds the largest value and re-scales all others according to it. IF the RGB value is negative, the code resets it to be zero. After the values are re-scaled, the code has certain logic corrections. If all of the colors are over saturated, rather than re-scale from the largest, the code resets all of the values to 255 so the light appears white. If all of the RGB values are equally and significantly low, the LED lights are all turned off.

IV. IMPLEMENTATION

To test and set up the system, we made a collection of colored objects. These objects, initially red, green and blue for threshold. Then orange, yellow, and pink objects were created for simulation and demonstration purposes. These objects were just plastic cups wrapped in colored tape.

V. BLOCK DIAGRAM

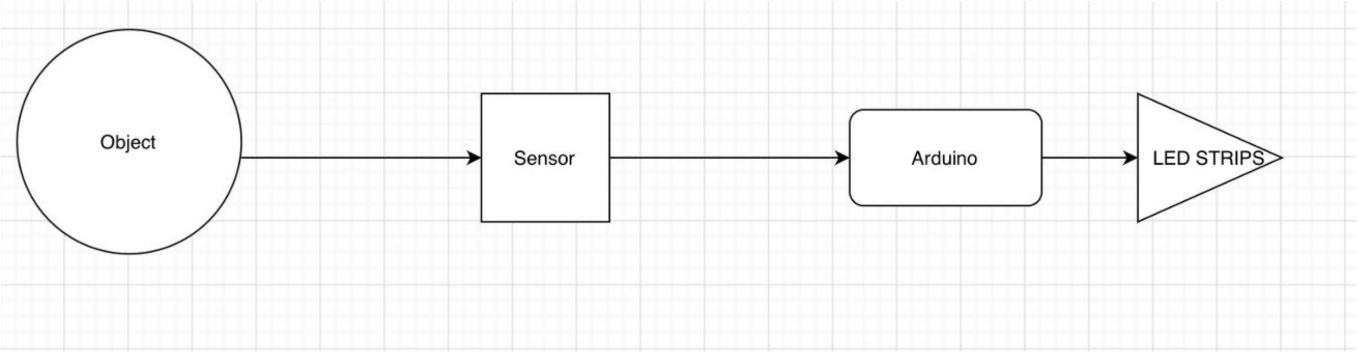

FIG. 1. Flow Chart

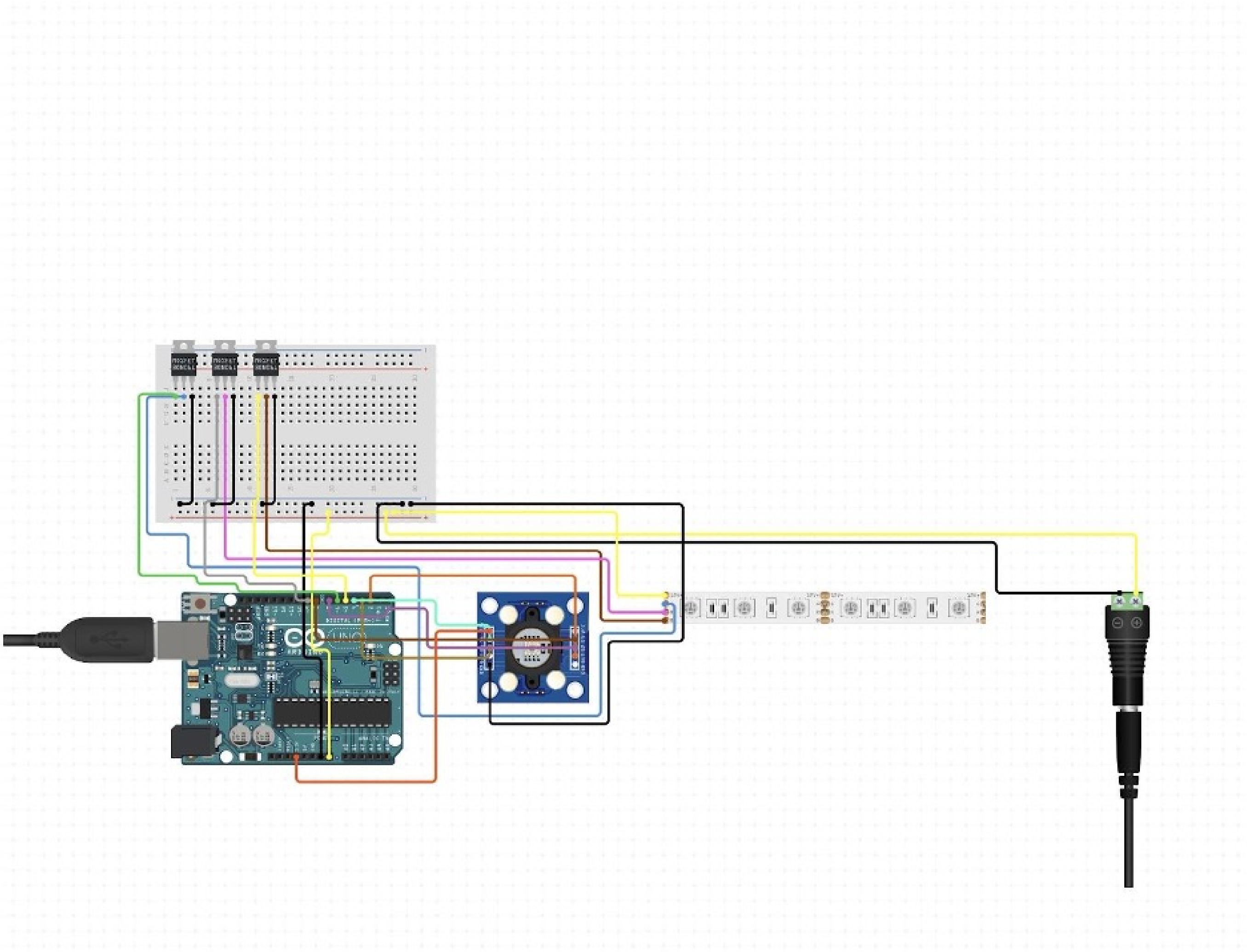

FIG. 2. wiring Schematic

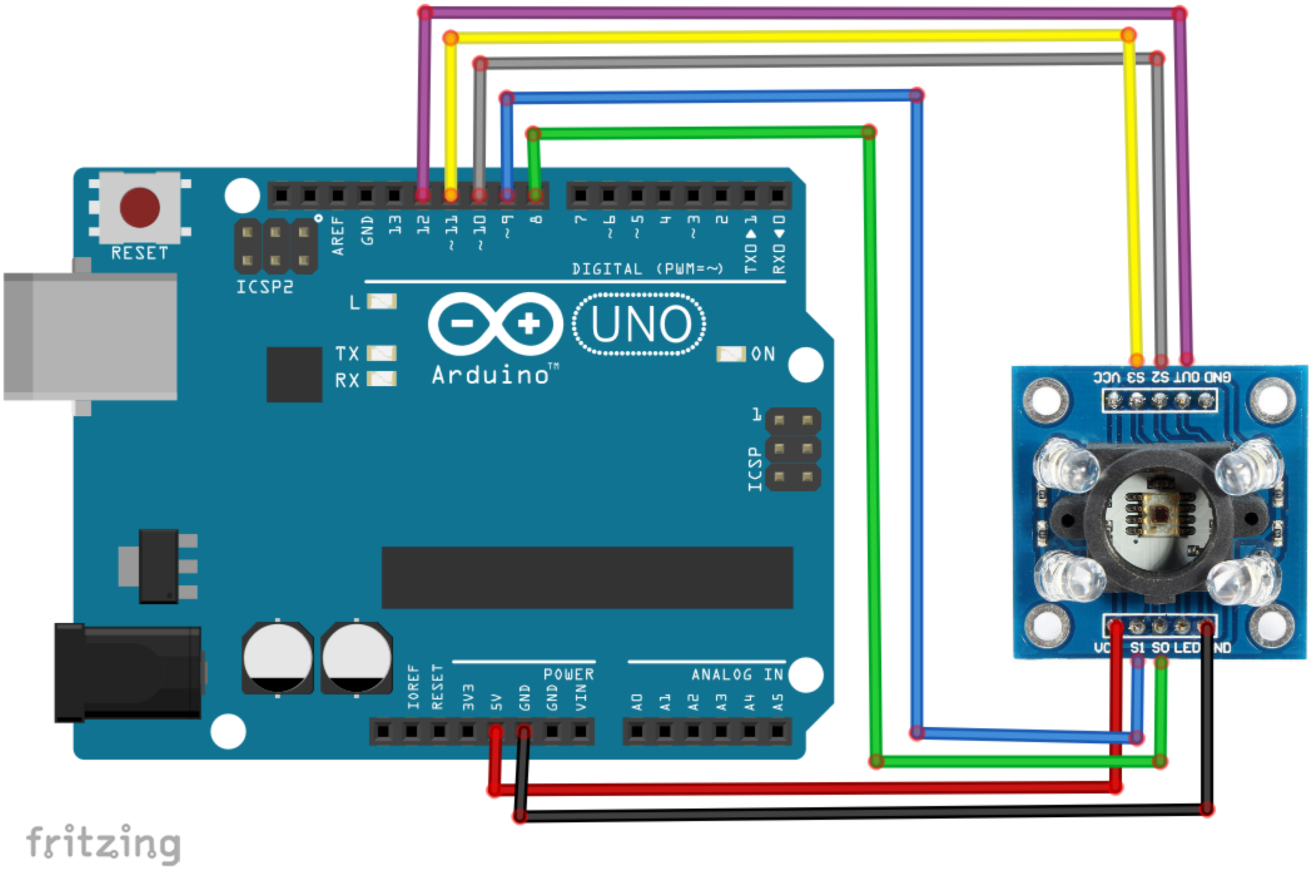

FIG. 3. wiring Schematic - Arduino to LED light strips