On the Limitations of Color Reproduction

Group 1

Siravid Chit-Arkhah, Rishi Mishra, Dominic Petruzzi, Colby Suppiger, Raymond Venneberg

University of Illinois at Urbana-Champaign

PHYS 398 DLP

Professor George Gollin

3 December 2021

Abstract

Any color humans perceive is the result of color receptors called cone cells in our eyes sending signals to the brain. Human eyes have three types of cone cells – each one sensitive to their own range of wavelengths within the visible spectrum. Color, as used in this paper, is the qualia[A] created by the brain due to stimulus from cones in the eye. When multiple wavelengths stimulate cone cells simultaneously, the brain blends them into a new color. Often, it is possible to create the same color given two different combinations of light. This phenomenon is called color metamerism and is a key concept when it comes to creating digital displays. The purpose of our experiment is to explore the phenomenon of color metamerism. Today, most digital displays span the standard red-green-blue

[A] See *§Appendix A: Definition of Terms* for definitions of words as used in this paper.

(sRGB)[A] color space[A] which is effective in replicating the majority of the human color vision gamut,[A] but it has its limitations. Using LEDs with non-conventional spectra[A], our goal was to produce a color that resides outside of the sRGB color space. In order to do this, software was written to simulate and predict the behavior of human eye responses to different light combinations. Our hardware was primarily used for taking data and displaying colors using LEDs. We were successful in creating different combinations of light resulting in the same color, however it is unclear if any of the produced colors lie outside of the sRGB color space.

Introduction

Purpose

The purpose of our project is to come up with a way for our LEDs to produce colors outside of the standard sRGB color space by covering more of the natural color gamut than the gamut of which the standard RGB LEDs can produce. Color gamut refers to a complete subset of colors, usually referring to the subset of colors that can be accurately represented in a given color space or certain display technology. If a color cannot be reproduced by a display, it is approximated to the nearest combination of wavelengths the LEDs can output. By adding another LED that outputs a different wavelength from the standard RGB LEDs, we can cover more colors which could possibly be reproduced in display technology. Figure 1 shows several different gamuts which are defined with LEDs of different wavelengths and excitation purities.[A] This would have applications in virtually any type of digital display i.e. TV screens, computer monitors, phones, etc. Ideally, we would be able to simulate the eye's response to a given spectrum of light produced by a set of LEDs. Knowing this, we would be able to plot out exactly what

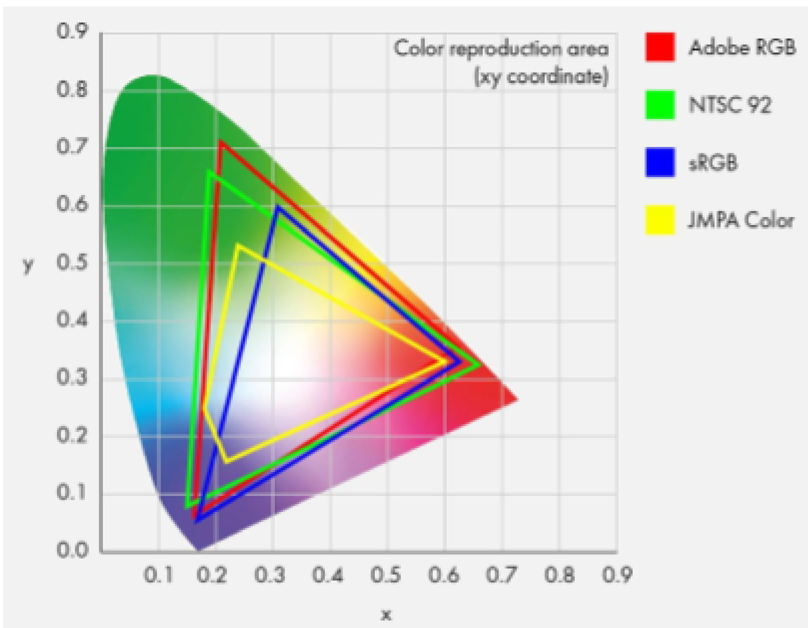

Figure 1: The xy chromaticity diagram with several different color spaces overlaid on it.

colors our different LEDs could produce and overlay it onto the sRGB color gamut which is the standard color space used to produce colors on modern displays. Using simulation software, we could find the LEDs with the most optimal range of colors and implement those combinations of LEDs into a digital device. Another goal of our project was to explore the way our eyes perceive color in more depth. The biological mechanisms behind sight are very complicated. An understanding of how the brain interprets color from both a single wavelength and a spectrum of wavelengths is necessary in approaching the topic of color metamerism. In order to achieve a wider color gamut, it is important to understand the process of how our brain interprets color so we can be able to test color reproduction within this new assumed-to-be-expanded gamut.

Our Initial Approach

To begin with, we wanted to come up with an algorithm that would take a desired single wavelength of light that we wanted to produce as an input and output the brightnesses that each LED needs to produce that color. The program was to be written in Python, and we go into detail about the math behind the algorithm used later on in the report. Next, we needed to come up with a way to measure our resulting combination of LED brightnesses. We used the Adafruit AS7262 color sensor to detect the wavelengths the LED combination was outputting. Finally, we needed a way to manipulate the brightness of each LED. Once we had these three processes figured out, we would be able to collect the data needed and analyze the results.

Background

Human Color Perception

The human visual system begins with the eye. Our eyes take in light from the visible spectrum and pass various signals along the optic nerves to the brain, where this information is interpreted to create one's visual perception of the world.

The human eye contains two types of visual receptor cells: cones and rods. Cones are responsible for color vision and are highly concentrated at the fovea, which lies at the center of vision. Light-sensitive receptor proteins in the cones, called photopsins, undergo chemical changes when they absorb light. This is the mechanism of photopic vision,[A] which is also called color vision. For rod cells, these proteins are called scotopsins.

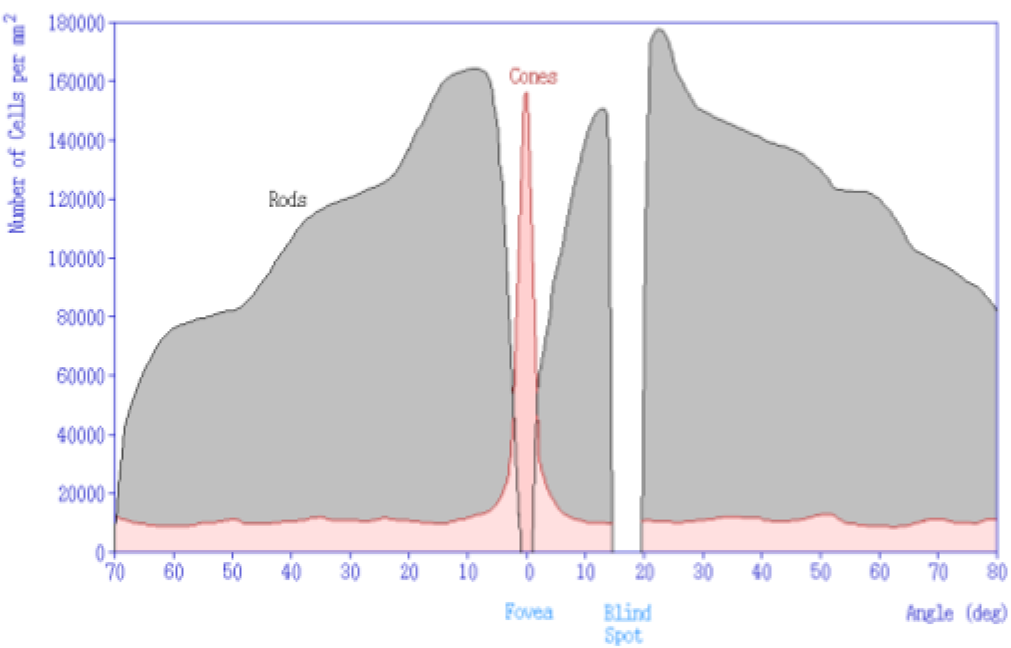

Figure 2: This distribution of cones and rods in the human eye. Cones have a large concentration near the center of vision with rods dispersed throughout the rest of the eye. The blind spot is where the optic nerve connects to the back of the retina.

Rods are primarily responsible for low-light vision and exist outside the concentration of cones. As rods are activated in dim environments and are insensitive to longer wavelengths of light, they play little role in color vision. Color perception is poor in low light conditions due to this and the fact that photopsins are inexcitable in such conditions. Figure 2 shows the distribution of cones and rods in the eye. As the amount of cones outside the fovea diminish, color perception in these areas becomes rather limited. However, due to the higher concentration of rods in these areas, there is better low-light perception. This is why it is easier to see stars in your peripheral vision rather than at the center of your vision.

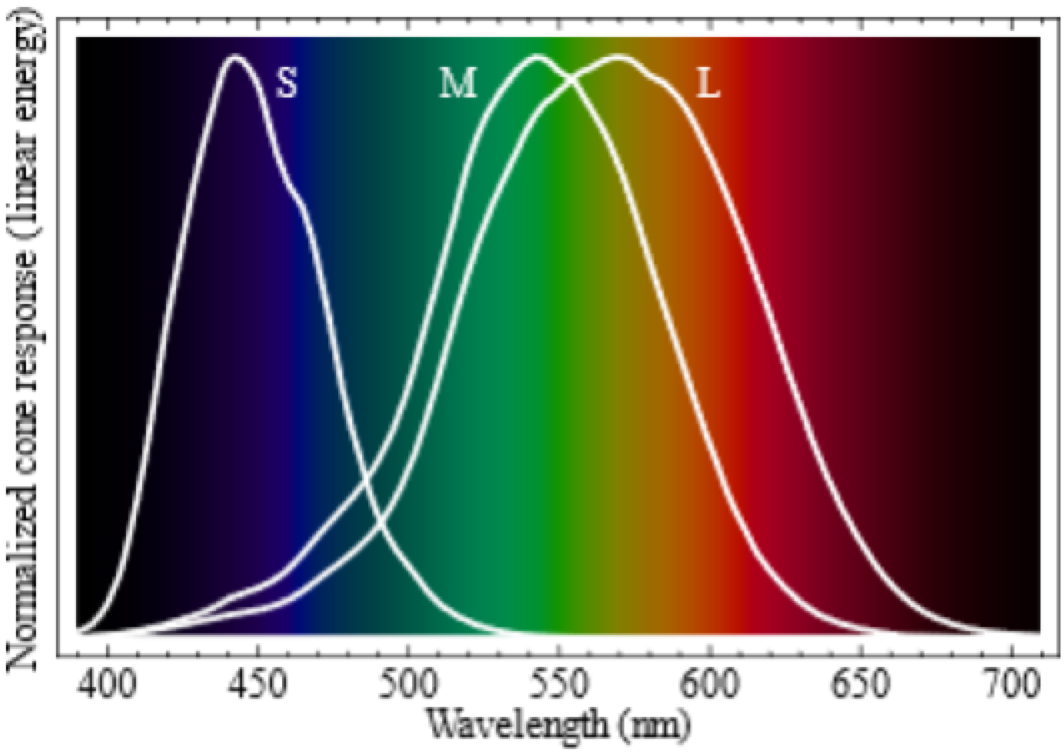

Figure 3: The normalized cone responses for the Short, Medium, and Long cones according to wavelength.

Cones are split into three categories: S, M, and L. These letters refer to cone cells that are sensitive to short, medium, and long wavelengths respectively. Figure 3 shows the normalized responses of each cone type to the visible spectrum. Short cones are primarily responsible for the perception of short wavelength light and blue colors. Medium and long cones cover the green and red ranges of the spectrum, and their response curves overlap heavily.

How these photoreceptors interact with the brain to create color vision is debated, but there are two main theories, the opponent-process theory and the trichromatic theory. The trichromatic theory was used as the main basis for this project because it is much easier to quantify the wavelengths we detect. This is necessary for the algorithms we use throughout this project. The trichromatic theory operates on the basis that the signals from the three cones are compared to determine the color and intensity of the light.

For trichromatism, the strength of the cone responses relative to each other provide the information to determine color and intensity, with multiple cone types necessary for distinguishing between different colors. A single type of cone would provide an intensity according to its response curve, with the greatest response of a given wavelength of light luminosity[A] occurring near the peak of that cone's response. One cone alone would not allow for the determination of color. Having two cone types like in people or animals with dichromatic vision, or otherwise known as dichromats, a comparison of response intensities can be used to determine color in addition to intensity. By adding a third cone type, humans can distinguish between millions of different colors.

A moderate response of the M cone could be the result of a medium intensity light near the middle of the visible spectrum or a high intensity light of a longer wavelength. As the peak response of the L cone is higher than the M cone and further towards the longer wavelengths, a light of a longer wavelength would stimulate the L cone more than the M cone, whereas a medium wavelength of light would provide similar responses from both the M and L cones. A comparison of the M and L cone responses would be able to determine the color and intensity of light for the spectrum range for which they overlap. Since a given wavelength can be represented in multiple ways using a combination of other wavelengths, you can create the same perceived color using different wavelengths and intensities of two or more emitters, such as the LEDs that were used for this project. This matching of colors is called metamerism, and the colors that create the matching set are called metamers. Figure 4 shows an example of the process by which this occurs. The single wavelength, through the cone sensitivities, creates a cone response that is perceived as a certain color. By stimulating the cone sensitivities via different wavelengths of certain intensities, the same cone response can be elicited which would create the perception of the same color.

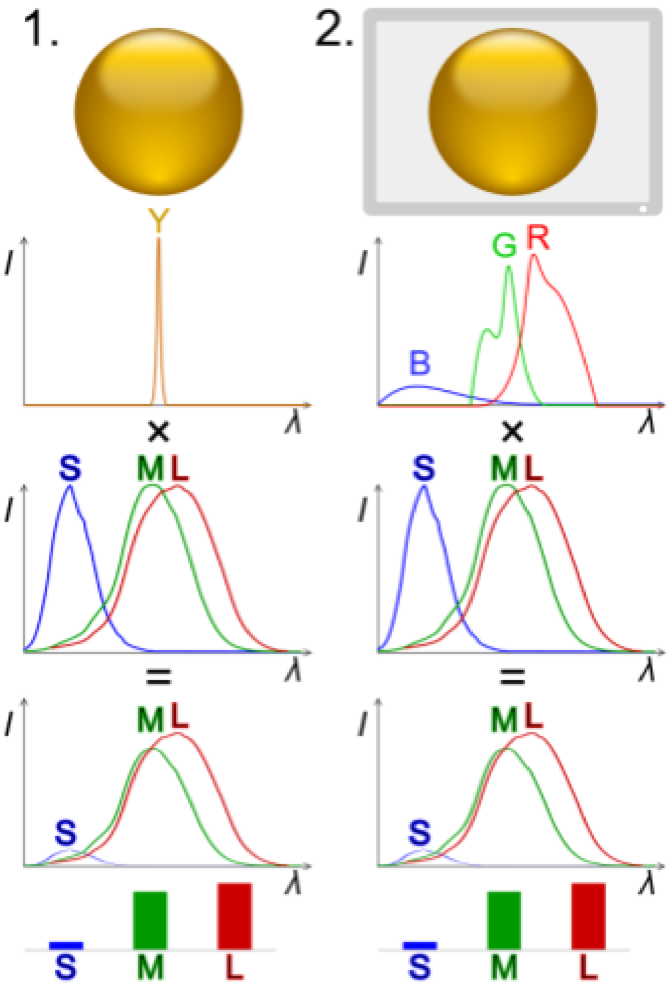

Figure 4: The reproduction of yellow light using red, green, and blue wavelengths.

Color Spaces

Several terms are used to characterize light and color. Some of the most important are hue[A], saturation[A] (similarly chroma[A], colorfulness[A], or excitation purity[A]), and luminance[A]. Black and white pictures are a pure luminant representation, showing only the brightness of light. Hue is the color of light corresponding to its counterpart on a color wheel, or rather whether a color is a red, green, yellow, orange, et cetera. Comparing an object and matching it to a color wheel would be an analysis of the object's hue. Saturation, chroma, colorfulness, and excitation purity are all representations of the intensity of color, or rather the difference between a white (or a neutral gray) and a given color. A pink and a red may have the same hue but differ in their saturation. Together, the hue and saturation are the chromaticity[A] of light. Chromaticity can be thought of as the characteristics of light without luminance, or rather what makes a color different from a given gray of the same brightness. This is what is represented in most diagrams of the CIE 1931 color space, with the x and y axes shown in Figure 7.

Figure 5: A color wheel showing many distinct hues.

Figure 6: A comparison of the Luminance and Chrominance of a given color image, demonstrating how they combine.

By comparing the intensities from all three cones, the color and intensity of light can be perceived by our brains to distinguish colors from one another. In order to represent the perceived color from a given wavelength of light, color matching functions[A] (CMFs) are used. These functions translate wavelengths of light through the cone responses to a certain color space. Color spaces are references which represent the possible colors observable by humans or the possible colors that can be produced by a display.

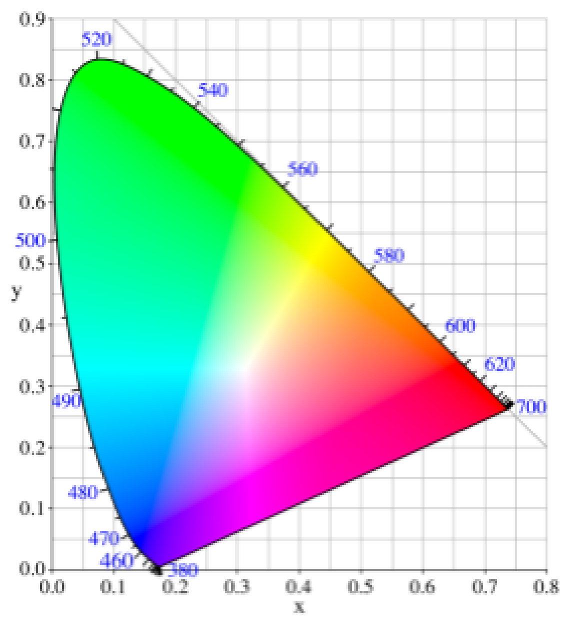

Figure 7: The CIE 1931 xy color space chromaticity diagram. The displayed colors are similar but not exact representations due to the limitations of displays and inks.

The most common and referenced color space is the 1931 Color Space designed by the Commission Internationale de l'éclairage (International Commission on Illumination, or CIE). This color space is defined in multiple representations: XYZ, xyY, and RGB. A representation of the CIE 1931 xyY color space is shown in Figure 7.

The CIE XYZ color space is supposed to represent all colors that are visible to a person with average eyesight. This is our basis for using tristimulus values as a representation of color as it uses a particular set of monochromatic light. In this model, Y is luminance, Z is close to the blue of the CIE RGB color space, and X is a mix of the CIE RGB curves that are nonnegative. If Y is luminance, the XZ plane will contain all possible chrominances[A], or the qualities of color at that luminance. The CIE xyY color space contains two normalized parameters derived from the XYZ components, x and y, created by the combination of the X, Y, and Z values. Here, x and y are known as chromaticity values.

The CIE RGB color space is distinguished by a set of monochromatic primary colors. This, however, limits the RGB gamut to a subset of the possible values in the CIE xy chromaticity diagram (the sRGB triangle in Figure 8). But we can go between the RGB and XYZ color spaces which makes it possible to connect RGB values to the xyY space.

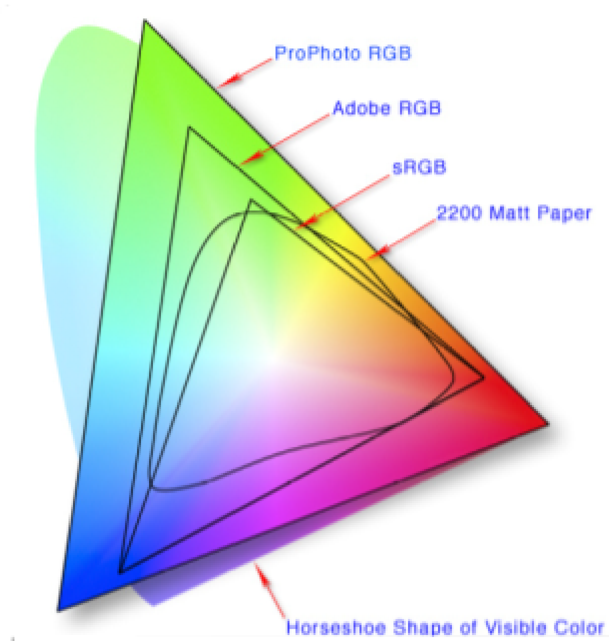

Figure 8: Comparison of different color spaces overlayed over the CIE 1931 xy chromaticity diagram. Note how none of the color spaces cover the entirety of the chromaticity diagram.

The CIE RGB color space covers the same area and colors as the CIE XYZ color space, but the values are defined as the amounts of red, green, and blue of a given color. The CIE LMS color space is created from the cone responses of the long, medium, and short cones in the eye. This defines colors by our perception of them and forms a basis from which the other color spaces are created. Going from one color space to another can be done via defined linear transformations and their inverses. This is essentially a change of basis in linear algebra using matrices. An example of this is shown in §*Procedures*.

Other color spaces are defined standards which describe the subset of colors that should be representable. The sRGB color space, which is the standard color space for the Internet, is the default color space if there is no other embedded color profile. Different image types and devices choose to adhere to one or more of these color spaces based on their capabilities and the purpose of the device itself. Color editing software with their standards use larger color spaces than the typical sRGB color space. Examples can be seen in Figure 8.

For cameras and monitors, the color gamuts in use are typically triangular. This is because usually only three types of sensors or emitters are used. This is compared to the color space standards where large overlaps in coverage are preferred to ensure accurate representation of the desired color ranges. Expanding the accurate color gamut of monitors is a driving factor in the development of high-quality monitors and television panels. Several aspects of color theory have been skipped or simplified for this overview. Due to this, please consider Figure 9.

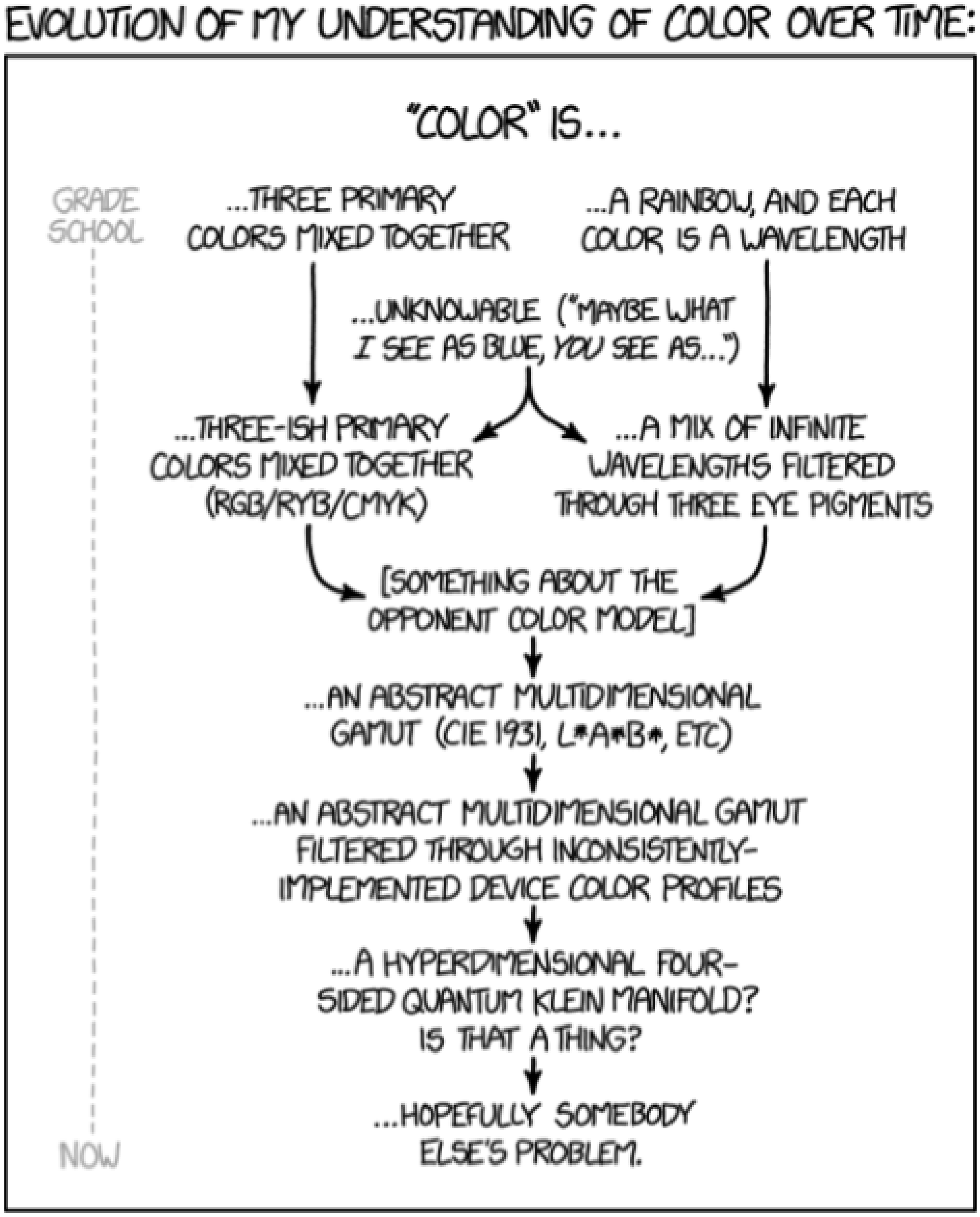

Figure 9: Color theory increases in complexity as more is learned. Courtesy of XKCD.com.

Hardware

Figure 10: Arduino Mega 2560

Arduino

An Arduino is an open-source electronics platform that conveniently combines the use of both hardware and software and can be used to conduct experiments and develop projects. The possible uses of an Arduino are virtually endless as it can be used for simple programs or for complex scientific instruments. The Arduino can take inputs such as a button click and can turn that click into any desired output like activating a motor or flipping a switch. The model we used is the Arduino MEGA 2560. This model has 54 digital input and output pins along with 16 analog inputs, 8KB of SRAM, 256KB of flash memory for storing code, and a clock speed of 16MHz.[1] The Arduino can be powered by a PC or by an external power supply. For example, in our project, we use a battery pack to supply power to the Arduino. To tell the Arduino what to do, a set of instructions can be sent to it using the Arduino's integrated development environment (IDE)[A] software where you can edit both setup and loop methods. Using the IDE, you can also direct information to post in a specific serial monitor. The Arduino is a core part of our printed circuit board (PCB) since it executes the code necessary for the device to run and communicates with each piece of hardware we use.

Light-Emitting Diodes (LEDs)

Most people are familiar with LEDs, however, it is still worth discussing the mechanisms behind their function. LEDs are a type of p-n junction, which is the boundary between two semiconductor materials – one p-type and one n-type. The p-type material has an excess amount of holes, which can be thought of as positive charged particles that can freely move around in the

[1] (*Arduino Mega 2560 REV3*)

material. The n-type material has an excess amount of electrons which can also move around freely in the material. Putting these two materials together results in the particles diffusing into the other material. This causes a "built-in electric field" within the material going in the opposite direction of the current flow. As a result, it is necessary to supply a voltage equal to this built-in electric field to allow current to flow through the diode to illuminate it. The reason behind the illumination of the diode lies within a concept called the band gap. In semiconducting material, there is a valence band energy level and a conduction band energy level. If an electron is stuck below the valence band, it is still attached to its host atom. If the electron wants to move around within the material, it must be excited up to the conduction band. For the excitement to happen, the electron must be supplied enough energy to overcome the difference in energy of the conduction band and valence band. When the electron falls from the conduction band back down to the valence band, the electron releases energy in the form of a photon. The energy and the color of the emitted photon is determined by the energy difference of the band gap. For our experiment, we used five LEDs, each with a different color: red, yellow, green, blue, and violet. Although our PCB had space for four LEDs, we were able to make metamers with just two or three of them illuminated at a given time. Ideally, our LEDs would be monochromatic, meaning they would emit only a single wavelength. In reality, our LEDs emitted a range of wavelengths that we needed to determine using the AS7262 color sensor.

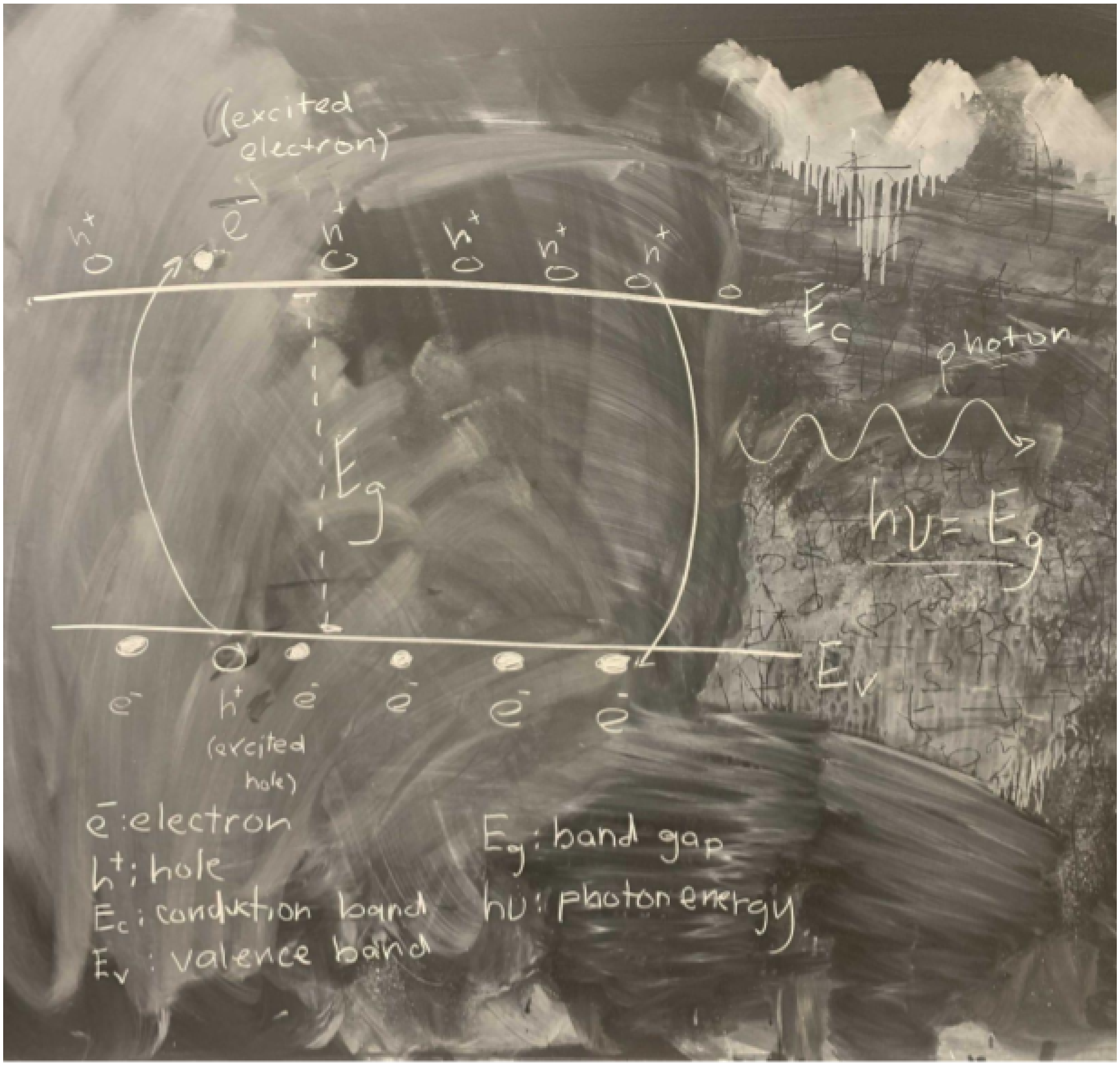

Figure 11: Band diagram of a semiconducting material. Electrons get excited up to the conduction band. Once the electron falls back down to the valence band, it releases energy as a photon.

MCP4728 DAC

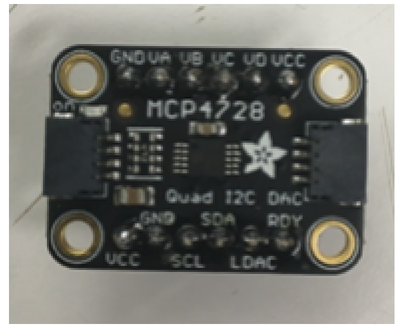

Figure 12: 4 channel DAC

The MCP4728 is a piece of hardware that houses four 12-bit digital to analog converters (DACs). The job of the MCP4728 is to supply voltage to our four LEDs. The DAC is capable of outputting between 0 and 5 volts, however the output function only accepts integer values between 0 and 4095 which meant that we had to use ratios to output a certain voltage. With the use of some transistors and variable resistors, we were able to vary the voltage going through each LED without having to reconstruct the voltage ratio for every LED in software. Now that both options are available to us, we can set our desired voltages through the LEDs by either adjusting the potentiometers or hard-coding the ratios into the Arduino IDE. Having both options is advantageous because it enables us to take data just as easily as it is to test the values that our code outputs.

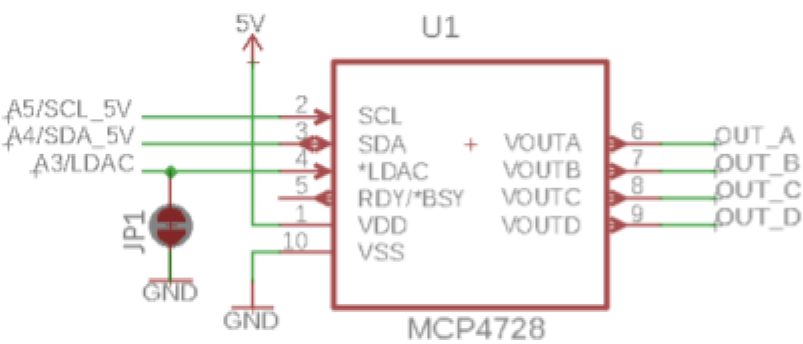

Figure 13: The MCP4728 DAC schematic

AS7262 Spectral Sensor

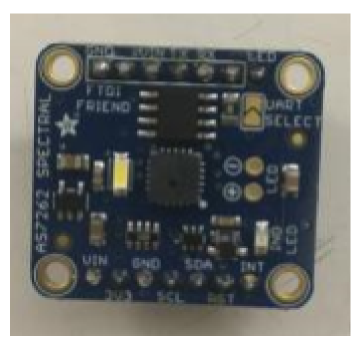
Figure 14: AS7262 spectral sensor

The AS7262 is a device that contains a color sensor with six channels along with some other functions that we did not use for this project. Each channel is sensitive to a particular intensity of light at one of the following wavelengths in nanometers: 450 nm (violet), 500 nm (blue), 550 nm (green), 570 nm (yellow), 600 nm (orange), or 650 nm (red). The channels themselves are made of silicon interference filters. These filters consist of multiple thin layers of silicon that have different indices of refraction. When light passes through these layers, wavelengths that are not the target wavelength are forced to undergo destructive interference, leaving only the light of the target wavelength. The intensity of this light is then measured, collected, and then transmitted to the Arduino.

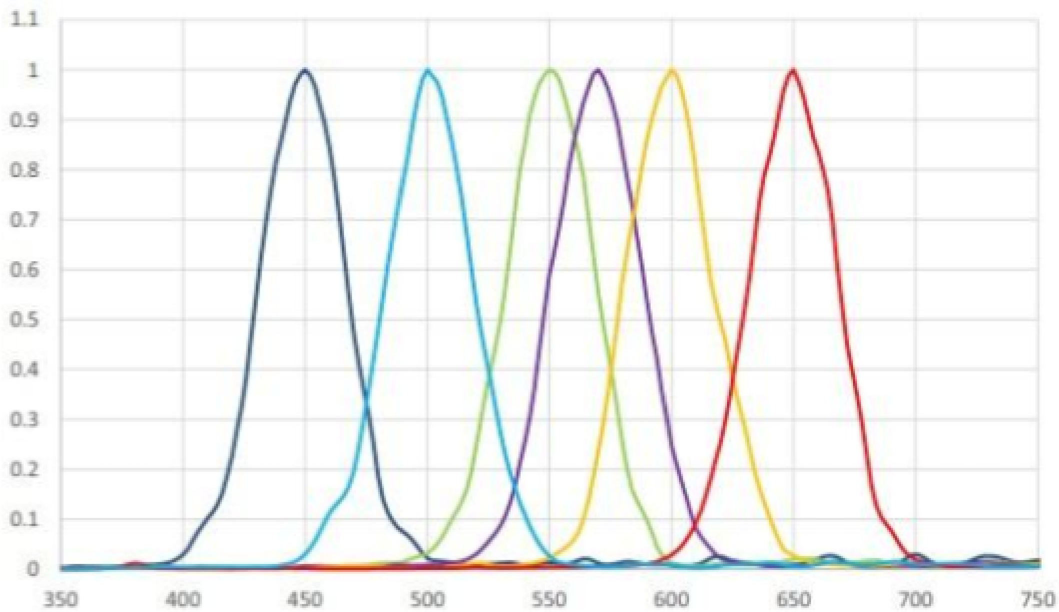

Figure 15: The spectral responses for each channel of the spectral sensor. Note how, although the maximum responses are at the stated wavelengths, the responses to wavelengths over 25 nm away from the stated wavelengths is still significant.

While this method of filtering light is effective, it has a few drawbacks. The first drawback is that, according to the spectral sensor's specification (spec) sheet, the typical accuracy of each channel is ±12% for response and ±5nm for wavelength.[2] Furthermore, each channel has a full width at half maximum (FWHM)[A] of 40 nm,[3] as seen in Figure 15. These large errors and tolerances create a significant amount of overlap between the channels. For example, monochromatic light at 525 nm is sensitive to both the 500 nm and 550 nm channels, leading the user to believe that both of those wavelengths are present when they are not. The second drawback is that lower-wavelength channels seem to have response "bumps" at higher wavelengths. This is especially prominent between the 650 nm and 750 nm curves. This could produce misleading data when an unknown color that consists of red and blue is tested.

Although the spectral sensor has drawbacks, we ultimately felt that it was the best device for our purposes. This sensor was an essential part of our project because it allowed us to objectively and quantitatively measure the colors we were producing, compared to the subjective and unreliable method of looking at and recording the colors ourselves. We used the data from the AS7262 in two ways: to construct the spectra of the LEDs, and to construct the spectra of the metamers.

Transistors

Figure 16: Transistor

The purpose of the transistors (in conjunction with the potentiometers) is to give us the ability to manually adjust the current going through the LEDs. Transistors have three connections: a base, an emitter, and a collector. By providing voltage to the base, a channel of

[2] (*AS7262 6-Channel Visible Spectral_ID Device with Electronic Shutter and Smart Interface* 2017)
[3] (*AS7262 6-Channel Visible Spectral_ID Device with Electronic Shutter and Smart Interface* 2017)

electrons is induced inside the material, creating the transistor. This allows current to travel from the collector to the emitter. The voltage difference between the base and the emitter is nearly constant at high currents – with the emitter being lower than the base – provided there is enough voltage passed through the base. In our case, this constant value is around 0.7 V. If a resistor is connected to ground on the emitter side of the transistor, this voltage and the voltage supplied to the base can easily be used to find a current through the circuit. The current supplied by the base is miniscule compared to the current supplied by the collector, meaning that the transistor can be used to easily adjust the current through the circuit on the collector side of the transistor. As shown in Figure 17, by putting the LED on the collector side, this property can be used to adjust the brightness directly.

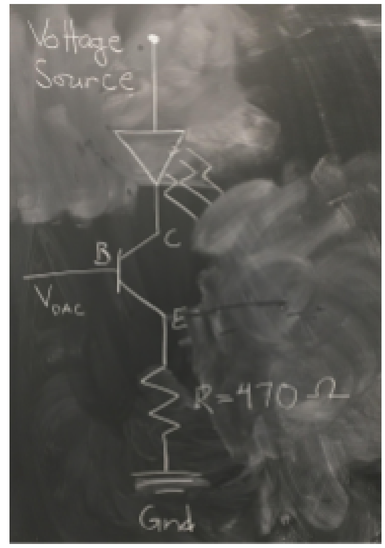

Figure 17: A section of our circuit. A voltage source (top) is connected to an LED and a transistor (center). The emitter wire of the transistor is connected to a resistor, which is connected to ground (bottom).

Fiber-Optic Cable and Diffuser

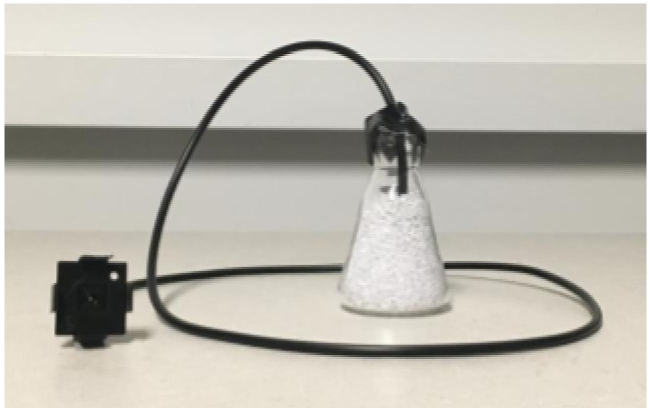

Figure 18: Fiber optic cable in diffuser

When viewing what color our LEDs produced, we needed to use a diffuser-- something that “mixes” our LEDs’ output wavelengths together. We created a diffuser by taking an Erlenmeyer flask and filling it with clear plastic beads. This would cause the light waves to bounce around until they all combined into one detectable spectrum. In order to connect the AS7262 to the diffuser, we used a fiber-optic cable. We buried one end of the fiber-optic cable into the beads in the Erlenmeyer flask and attached the other to a clip that we 3D printed to attach to the AS7262. While the color detected by the AS7262 did depend on the exact position of the buried end of the fiber-optic cable and the displaced distribution of the beads, we felt that the small fluctuations of color we saw due to the slight movement of the fiber-optic cable inside the flask was negligible since the color was homogeneous within the region the cable was in.

Liquid Crystal Display (LCD)

Figure 19: LCD attached to the PCB. The LCD is displaying the DAC counts of the four LEDs: red (0), green (1), blue (2), and yellow (3).

Our printed circuit board also included an LCD to be attached. This display uses a dot matrix to display two rows of 16 characters each. A dot matrix consists of a grid of small squares that can be illuminated individually to display characters. This is in contrast to a seven-segment display, in which characters are broken down into horizontal and vertical lines (think a squared off "8") which are illuminated individually. We used the LCD to display the voltages of the LEDs (in number of counts) in real time as we adjusted the potentiometers. This helped us replicate and alter LED intensities quickly without needing to stop and change the Arduino code.

Other Hardware Components

In addition to other components above, our printed circuit board included resistors, potentiometers (or trimpots), capacitors, a battery pack, a keypad, and a MicroSD port. The purpose of the potentiometers and resistors was to allow us to control and vary the voltage through the LEDs as mentioned in §§*Light-Emitting Diodes*. The battery pack and capacitors were used to supply the circuit components with voltage and current. The keypad and MicroSD port were included to facilitate the input of parameters and output of data; however, we found it much more efficient to handle this input/output (I/O) in the Arduino code.

Connections Between Hardware Components

Each of our LEDs' shorter legs (where the voltage should be lower in the circuit) was connected to the collector of each negative-positive-negative (NPN) transistor; the longer legs were connected to the positive 5 V line. Connected to the bases of each transistors were the four output lines from our DAC. The emitter sides of the transistors were connected to resistors which then completed the circuit on the negative line. The LEDs, however, needed a larger voltage, so we attached a wire connecting them directly to the batteries, allowing them a potential lift up to almost 7.5 V. The DAC itself was connected to the main circuit to deliver power and data to the Arduino and the LEDs. The AS7262 sensor was connected to the main board where it was fed a power and data cable. The diffuser was then attached to the sensor via a fiber-optic cable.

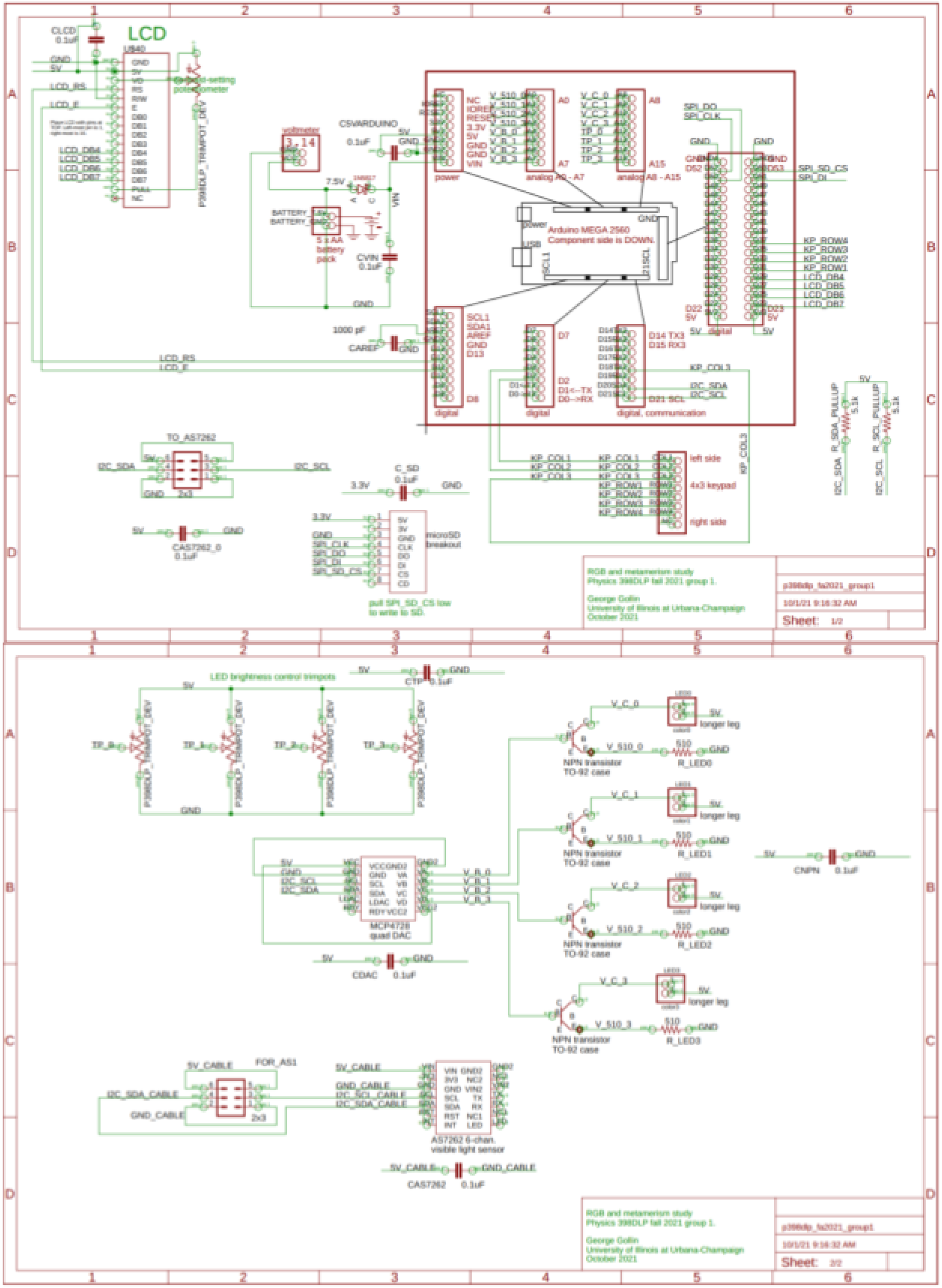

Figure 20: Schematic of the circuit on our PCB.

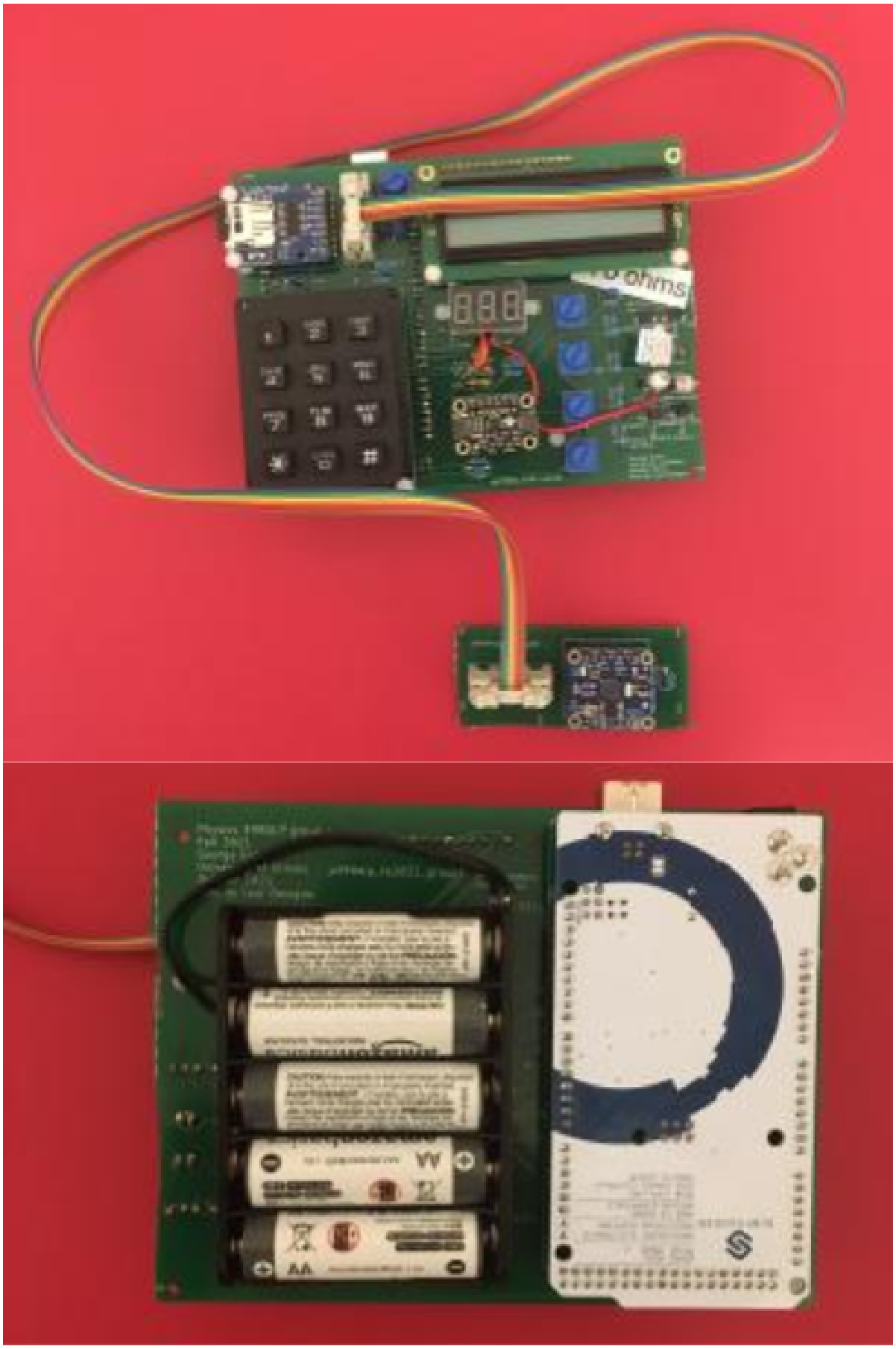

Figure 21: PCB modified to connect transistor collectors directly to the battery pack. Note the red wire running from the center of the board (connected to the battery pack) to the LEDs.

Software

Python

Python is an interpreted, object-oriented, high-level programming language that, among other things, allows users to write software to conduct data analysis. Using the NumPy, pandas, and matplotlib libraries, we were able to utilize Python to sort and analyze data pulled from the internet and data that we collected. Using NumPy matrices, we were able to create an algorithm to give us normalized values for how bright each LED needs to be to produce a desired wavelength. Essentially, Python was the backbone of all our calculations and data analysis for this project.

Arduino Code

The Arduino IDE utilizes a language similar to C. This language was used to communicate between the Arduino and the components on the PCB. We used several libraries, each one being necessary to communicate between the Arduino and each component on the board. To view our collected data, the Arduino IDE displays the data output on the serial monitor.[A] Here, we could have utilized the SD card and its reader to have the Arduino print the data to it but we found it easier just to copy the data that output onto the serial monitor and copy it into a text file instead.

Procedures

Getting Cone Responses

To start, we needed to know the exact response of each cone to all wavelengths in the visible spectrum. This has already been studied,[4] so we used data we found online for our project. A sample of this data is shown in Figure 22.

	A	B	C	D
1	Wavelength (nm)	Long Cone Response	Medium Cone Response	Short Cone Response
2	391	5.03E-04	4.48E-04	1.15E-02
3	430	2.82E-02	3.95E-02	8.03E-01
4	480	1.40E-01	2.36E-01	3.90E-01
5	530	7.71E-01	9.36E-01	1.26E-02
6	580	9.69E-01	6.53E-01	1.09E-04
7	630	4.01E-01	6.21E-02	0
8	680	2.54E-02	1.64E-03	0
9	730	6.39E-04	4.16E-05	0
10	780	2.07E-05	1.63E-06	0
11	830	9.74E-07	9.53E-08	0

Figure 22: Sample of reference data found online. Each entry has a wavelength measured in nm (Column A) and the responses of the long (Column B), medium (Column C) and short (Column D) cones when light with that wavelength is shined in an eye.

The four columns of Figure 22 represent the four components for each data point. Column A is the wavelength of the light, measured in nanometers. Columns B, C, and D are the responses in each of the three cones in the human eye; these columns represent the response in the cones sensitive to long, medium, and short wavelengths, respectively. These numbers have arbitrary units and are normalized. In general, the long cone is most sensitive to red light, the medium cone to green light, and the small cone to blue light. The data covers wavelengths from 390 nm through to 830 nm with a resolution of 0.1 nm, covering the visible light spectrum.

[4] (Color and Vision Research Laboratory, *Cone Fundamentals*)

Converting Cone Responses into Intensity Values

To convert the response arrays into RGB arrays, we need to apply a couple of transforms to each of the arrays. We create an array called Γ_{cr} which contains the responses of the long, medium, and short cones from a given spectra:

$$\Gamma_{cr} = \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

We can multiply this matrix by a series of transform matrices to convert the cone response into an RGB array. It is worth noting that this process will only result in one of the multiple possible RGB values that can be created from the response matrix. First, we need to transform the cone response into the XYZ color space. This is done by multiplying our cone response matrix by:

$$T_{LMS->XYZ} = \begin{bmatrix} 1.91 & -1.11 & 0.201 \\ 0.371 & 0.629 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}$$

This transform assumes the LEDs emit a single wavelength of light when, in reality, they emit a spectrum of light. To adjust for this, we can model the LED spectra as thousands of monochromatic wavelengths, each with their own cone response matrix. To get a single effective cone response matrix for the LED, we take the weighted average of all the matrices with the weight of each matrix determined by the relative intensity of the wavelength in the LED's spectra. Second, we need to transform our XYZ color space values into RGB values. This is done by multiplying our values with another transform matrix:

$$T_{XYZ->RGB} = \begin{bmatrix} 0.418 & -0.159 & -0.0828 \\ -0.0912 & 0.252 & 0.0157 \\ 0.000921 & -0.00255 & 0.179 \end{bmatrix}$$

Once we have done the transforms we have arrived at our RGB values.

$$\Gamma_{RGB} = T_{XYZ->RGB} \times T_{LMS->XYZ} \times \Gamma_{cr}$$

$$\Gamma_{RGB} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Suppose we do this for our three LEDs, their corresponding cone responses, and the wavelength of our desired color. Now, we have four RGB matrices – one for each of our LEDs and one for our desired wavelength.

$$\Gamma_{RGB_\lambda} = \begin{bmatrix} R_\lambda \\ G_\lambda \\ B_\lambda \end{bmatrix} \Gamma_{RGB_{rLED}} = \begin{bmatrix} R_{rLED} \\ G_{rLED} \\ B_{rLED} \end{bmatrix} \Gamma_{RGB_{gLED}} = \begin{bmatrix} R_{gLED} \\ G_{gLED} \\ B_{gLED} \end{bmatrix} \Gamma_{RGB_{bLED}} = \begin{bmatrix} R_{bLED} \\ G_{bLED} \\ B_{bLED} \end{bmatrix}$$

We can now set up a linear system with our four matrices.

Once we have this linear system, we can use the Python library, NumPy, to solve for our intensity values. Finally, we arrive at our intensity matrix.

$$\begin{bmatrix} R_\lambda \\ G_\lambda \\ B_\lambda \end{bmatrix} = I_r \begin{bmatrix} R_{rLED} \\ G_{rLED} \\ B_{rLED} \end{bmatrix} + I_g \begin{bmatrix} R_{gLED} \\ G_{gLED} \\ B_{gLED} \end{bmatrix} + I_b \begin{bmatrix} R_{bLED} \\ G_{bLED} \\ B_{bLED} \end{bmatrix}$$

$$\begin{bmatrix} R_\lambda \\ G_\lambda \\ B_\lambda \end{bmatrix} = \begin{bmatrix} R_{rLED} & R_{gLED} & R_{bLED} \\ G_{rLED} & G_{gLED} & G_{bLED} \\ B_{rLED} & B_{gLED} & B_{bLED} \end{bmatrix} \begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix}$$

$$I = \begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix}$$

We then use NumPy to normalize the matrix to arrive at our final intensity values for each of the LEDs to produce our desired color.

It is worth mentioning that these values are arbitrary and range between 0 and 1. For example if our matrix is:

$$I = \begin{bmatrix} 0.5 \\ 0.333 \\ 1 \end{bmatrix},$$

our red LED needs to be half the brightness of our blue LED, and our green LED needs to be one-third the brightness of our blue LED.

Pre-Processing

Metamers can be found by taking a set of LED intensities selected at random and determining the cone responses caused by that set of LEDs' intensities. Then, in hardware, we can change the LEDs to ones of different spectra and pass those cone responses through the code reviewed above. This results in a set of intensities for the new LEDs that will blend into the same color as the old combination. Naturally, the ratio of each intensity to the others is the determining factor for the color while the actual magnitude of each is unimportant given that no other light is present in the data. This means that the intensities found can be safely normalized for ease in processing.

The intensity of light from each LED is dependent on the power supplied to the LED. The power running through an object in a circuit is equal to the product of the voltage across that object and the current running through it. We found that the voltage across the diode is equivalent to the numerical value of its band gap when expressed in electronvolts, to within a reasonable amount of error. The current running through the diode is approximately equal to the current running through the resistor on the other side of the transistor. With the bandgaps of each LED calculated and assigned to constants in Python, the power produced by the LED can be found by the equation:

$$P = \frac{V_{DAC} - 0.7}{Resistance} * Bandgap \text{ (eq. 1)}$$

which can be rearranged to find the voltage required from the DAC for any given power:

$$V_{DAC} = \frac{Resistance * P}{Bandgap} + 0.7 \text{ (eq. 2)}$$

In order to get the photon count emitted from the LED for a certain power, the following equation can be used:

$$photon\ count\ per\ second = \frac{power}{bandgap}\ (eq.\ 3)$$

Once the required relative intensities are found, the proportional amount of power can be found using Equation 3. The LED with the lowest required power is then set at a fixed value. If the lowest normalized proportion is less than 0.15, the amount of power put through this LED is set at zero, and the next smallest required power is set as the new lowest proportion. At this point, the LED with the lowest required power will have its power set as the amount of power resulting from 0.74 V supplied to the base of the transistor from the DAC (an output of 600 out of 4095). This magnitude of power is then used to find complete photon counts from the respective LED, which is then used to find the photon counts from the other LEDs. These counts can then be used to find their required DAC outputs using Equation 1 and Equation 2. Note that the DAC output of the dimmest LED was selected to be above the minimum DAC output that would have resulted in non-zero power (as 0.74 V > 0.7 V).

Displaying Color Based on Numbers Outputted by Intensity Program

The DAC is capable of outputting between 0 and 5 volts, but the output function accepts integer values between 0 and 4095. Once the correct voltage was determined by the aforementioned Python program, it was divided by 5 to get a ratio of the maximum output from the DAC to be outputted. This number was then multiplied by 4095, rounded to the nearest integer, which was then passed through the output function. This method supplied the correct voltage to the transistor in order to pull current through the diode. It was also used for when metamers were determined from the code.

When metamers were determined by eye, the brightness of the LEDs were not directly controlled by code but rather by the trimpots connected to the PCB. As shown in Figure 23, the translation from trimpot resistance to DAC output was fairly simple; varying the resistance across the trimpot directly affected the voltage read by the analog input channels in the Arduino.

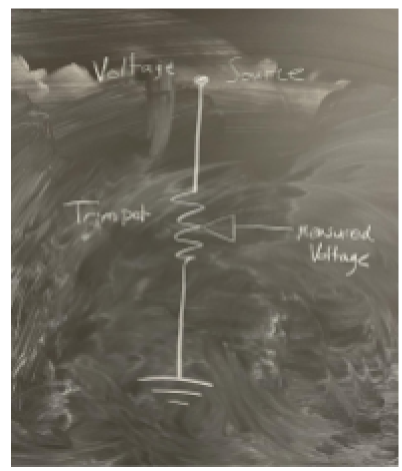

Figure 23: Diagram of circuit including the trimpot for adjusting voltage by hand.

This was read as a number between 0 and 1023. The ratio of the input with 1023 as the denominator was then multiplied by the maximum value the DAC could output – 4095. It was further passed through the output function for the DAC. Creating this hand-adjustable method of control resulted in an easy way to modify the LED brightnesses.

Determining LED Spectra

An LED emits a spectrum of light with different wavelengths which can have different intensities. These spectra follow a Gaussian distribution centered around the wavelength at the greatest intensity, the peak wavelength. Although information on spectra and peak wavelengths is usually provided in the LEDs' specification sheets, we were unable to find sufficiently detailed information about the spectra of all of our LEDs. In order to ensure that our spectra were consistent for all of the LEDs, we used the spectrometer to measure the spectra of all of our LEDs instead.

In order to create the spectrum for an LED, we first found the intensity of light emitted at each of the spectrometer's set wavelengths. We placed the LEDs in our PCB and adjusted the potentiometers so that one LED was at the maximum brightness while all the other LEDs were off. Then, we held one end of the optical fiber roughly 1 cm above the shining LED and connected the other end of the optical fiber to the spectrometer. In order to minimize noise, this was done in a dark room. We then ran our Arduino program that outputted the intensity of light (measured in counts) at each of the spectrometer's specific wavelengths several times per second. We ran this program for about 15 seconds for each LED. Then, we inputted the intensity data into a Python program that outputted the average intensity at each wavelength (Table 1). We used these average intensities to create a function that described the spectrum of light emitted by the LEDs.

Wavelength (nm)	Red LED	Yellow LED	Green LED	Blue LED
450	1161	419	29	31724
500	1294	421	750	17163
550	901	12944	2072	153
570	4172	41443	1033	288
600	3118	48476	153	86
650	48742	958	22	294

Table 1: Displays the intensities of each LED in spectrometer counts at each wavelength measured by the spectrometer. For a graphical representation, see Figure 24. RMS jitter for each channel was calculated to be less than 1% of the measured values.

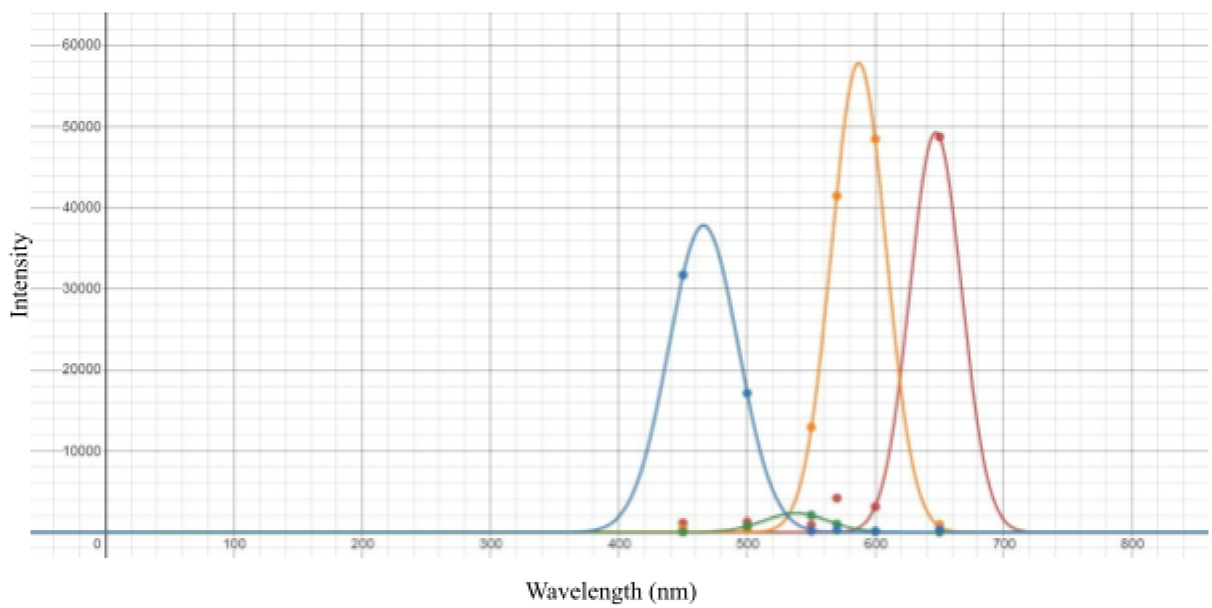

Figure 24: Intensity (measured in spectrometer counts) vs. wavelength (measured in nm). Points represent raw values measured, while Gaussian curves are of best fit.

We fit the average intensities to a Gaussian curve using Desmos. First, we expressed the average count numbers as fractions of the maximum intensity that the spectrometer could measure (2^{16}-1 counts). Then, we fit each set of fractions to the function,

$$I = ae^{-\frac{1}{b}(x-\lambda)^2}$$

,

where I is the count number at wavelength x, and b is a coefficient that determines the width of the Gaussian. In Desmos, we defined λ as a constant and used sliders to adjust its value until the coefficient of determination (R^2) was maximized. This value of λ was taken to be the peak wavelength. Desmos also outputted ‘a,’ the amplitude of the Gaussian. Since we were able to find the peak intensity of all of the LEDs from the spec sheets, we changed a to the peak intensity after determining λ. Using Desmos, we were able to turn our spectral data into a function of wavelength.

Determining LED Peak Intensities

The spec sheet of each LED contained information about its peak (or maximum) intensity (Table 2). As seen in the table, the peak intensities vary greatly. This posed a problem: if we tried to mix the light from two LEDs with very different peak intensities, the light from the brighter LED would “drown out” the light from the dimmer LED. In order to mitigate this, we soldered three red LEDs together to bring the total peak intensity closer to that of the other LEDs, hence the multiplication in the Red column in Table 2.

	Red[5]	Yellow[6]	Green[7]	Blue[8]
Peak Intensity (mcd)	1500 × 3 = 4500	18000	8000	6000

Table 2: Displays the peak intensity of each LED. Superscripts direct to the source of this information – the spec sheets of the LEDs.

Data Collection

The project heavily utilized software in both data collection and device control. Software for data collection was mainly written in C in the Arduino IDE while data processing software was written in Python. This allowed for easy use of the hardware connected to the Arduino.

LED intensities were determined using one of two methods: manual adjustment and code output. For manual adjustment, we used two identical PCBs and sets of LEDs. We would place the diffuser over each LED set and look through the fiber optic cable to determine the resultant color. Then, we would turn the dials on the trimpots to make the resultant color match the resultant color on the other PCB more closely. Once we could no longer distinguish the colors apart, we recorded the two sets of DAC counts displayed on the LCDs. For the code output, we would run the computer programs described in previous sections to find two sets of DAC counts that produced similar colors. We then recorded the sets of DAC counts.

[5] (Adafruit Industries, *Super Bright Red 5mm LED (25 pack)*)

[6] (Adafruit Industries, *Super Bright Yellow 5mm LED (25 pack)*)

[7] (Adafruit Industries, *Super Bright Green 5mm LED (25 pack)*)

[8] (Adafruit Industries, *Super Bright Blue 5mm LED (25 pack)*)

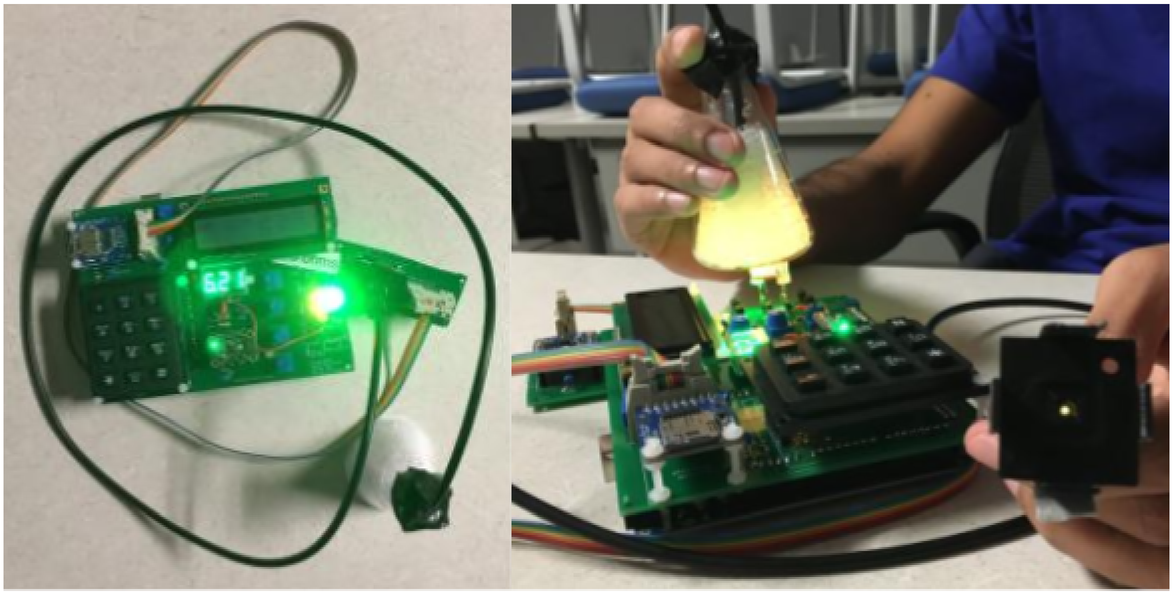

Figure 25: Our experimental setup for data collection (left). Light traveling through the fiber optic cable (right). Compare the light coming from the diffuser and the fiber-optic cable.

Once the LED intensities were set, we began to take spectrum data. This process is almost identical to the process we used to find the spectra of the LEDs. With multiple LEDs, however, we needed to put the light-collecting end of the optical fiber into the diffuser. We buried the end of the fiber roughly 2 cm into the beads. Then we held the whole diffuser about 1 cm above the LEDs. Data was collected about 10 times per second. The raw values outputted by the light sensor were then collected in a text file. These values were then passed into Python code which parsed and averaged the values taken during the data collection period for each of the measured wavelengths. This resulted in six values of intensity corresponding to each wavelength discerned by the light sensor.

Using the same Arduino and Python programs that we used when finding the spectra of the individual LEDs, we obtained the average intensities (in counts) for each of the spectrometer's wavelengths. The diffuser mixed the light from each LED which allowed us to collect data from a set of LEDs to produce a set of average intensities.

Post-Processing

The cone response of the spectrum created from the LEDs could be approximated by the sum of the colorimeter responses due to the intensities at each measured wavelength. These responses were computed via a Python program that was written and used to search the comma-separated values (CSV) file containing cone responses due to wavelengths for each of the measured wavelengths. The responses for the red, green, and blue cones were determined by a weighted average of the found responses where the whites were determined by the proportion of the intensity of each wavelength compared to the sum of all of the intensities. While the value of each individual number is important in determining the total brightness of the emitted light, the ratios between the numbers determine the spectrum and, thus, the color displayed. This means that the responses could be normalized without compromising the color they represent. From reading the previous sections, you could see that normalization was done to maintain consistency throughout.

Results

By using our Python code and manually adjusting the intensities of the LEDs using the potentiometers, we were able to find three different metamers: purple, yellow, and orange. These metamers were found to be the same by multiple subjective accounts. ‘DAC Counts’ are the values used for the LED combo that was given a voltage for that specific test for color reproduction. The cone responses were normalized for data usage purposes and clearer readings.

The following format was used to record metamer data:

Metamer Color Name					
LED Combo	DAC Counts	Spectrum	Cone response <R, G, B>	xy Color Space values	Outside sRGB?
The combination of LEDs used to produce the metamer.	The DAC output (between 0 and 1023) that was sent to the transistor controlling the specific LED.	The spectrum of the diffused light as measured by the colorimeter with responses from the six wavelengths recorded as a set in [brackets].	The normalized vector containing the red, green, and blue cone responses in that order for the spectrum provided.	The location in the xy color space.	Whether or not the measured point lies outside of the sRGB triangle in the xy color space.

The ‘Difference’ row is either a qualifier to determine if the combinations of LEDs differed in a significant manner under each of the columns. Where appropriate, a numerical percent difference is provided.

Purple					
LED Combo	DAC Counts	Spectrum	Cone response <R, G, B>	xy Color Space values	Outside sRGB?
Red Blue	799 911	[344, 111, 6, 7, 47, 60]	<0.201, 0.191, 0.608>	<0.284,0.195>	No
Blue Yellow	711 883	[107, 46, 5, 23, 17, 1]	<0.267, 0.257, 0.475>	<0.320, 0.260>	No
Difference	Yes	Significantly Different (GOOD)	<28.2, 29.5, 24.6> (in terms of %)	<11.92, 28.57> (in terms of %)	No

Table 3: Properties of the purple metamers.

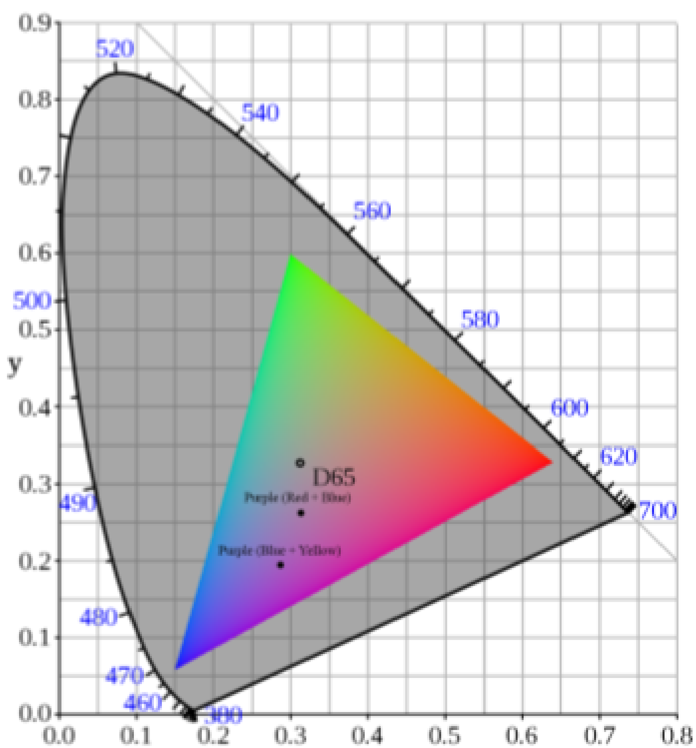

Figure 26: Purple metamers on the xy color space. The triangle is the standard rgb color space, and the curved shape is the range of human color vision.

Yellow					
LED Combo	DAC Counts	Spectrum	Cone response <R, G, B>	xy Color Space values	Outside sRGB?
Yellow	3500	[419, 421, 12944, 41443, 48476, 958]	<0.598,0.398,0.003>	<0.700,0.473>	Yes
Red Green	2255 1119	[1, 25, 40, 22, 41, 56]	<0.561, 0.418, 0.021>	<0.611,0.471>	Yes
Difference	Yes	Significant	<6.38, 4.90, 150> (in terms of %)	<13.6,0.424> (in terms of %)	No

Table 4: Properties of the yellow metamers.

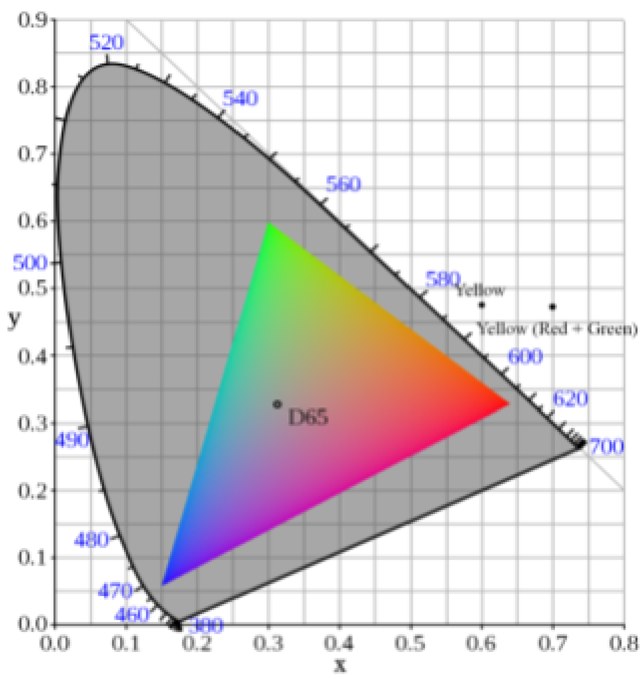

Figure 27: Yellow metamers on the xy color space. The triangle is the standard rgb color space, and the curved shape is the range of human color vision.

Orange					
LED Combo	DAC Counts	Spectrum	Cone response <R, G, B>	xy Color Space values	Outside sRGB?
Red Green	1227 535	[1,6,15,12,69,49]	<0.640, 0.348, 0.012>	<0.838, 0.456>	Yes
Red Yellow	703 947	[0,1,9,40,62,7]	<0.618, 0.381, 0.001>	<0.751, 0.463>	Yes
Difference	Yes	High	<3.498, 9.053, 169> (in terms of %)	<10.95, 1.52> (in terms of %)	No

Table 5: Properties of the orange metamers.

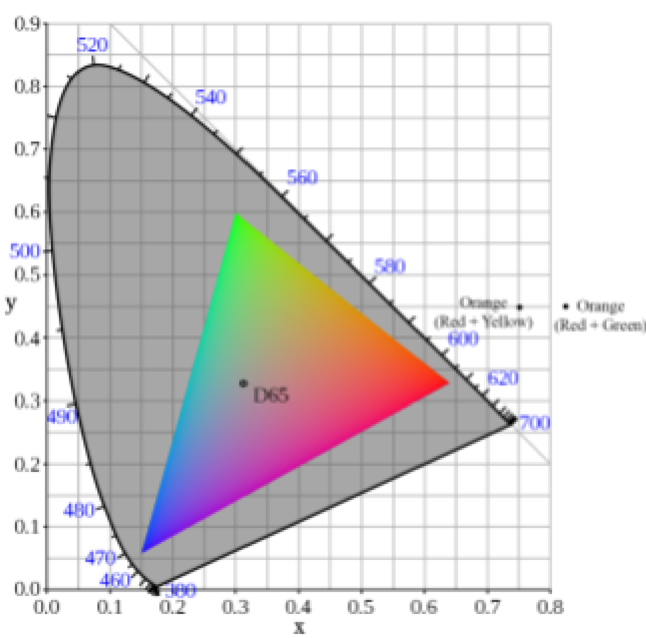

Figure 28: Orange metamers on the xy color space. The triangle is the standard rgb color space, and the curved shape is the range of human color vision.

Discussion

A few problems arose when constructing the circuit for use in data collection. Aside from minor problems such as connectivity issues and improper wiring, one glaring issue presented itself: transistor saturation. Initially, the maximum voltage supplying the collector side of the transistor through the LED was 5 V. In our circuit, when the DAC supplied a full 5 V (or nearly 5 V) to the base of the transistor, there was insufficient voltage difference between the base and the collector. This meant that no current flowed through the circuit, and the LED would not turn on. When this happens, the transistor is said to be saturated. Thus, the modeled behavior and actual behavior of the power through the LED conflicted. This limited the operating range of the LED enough to create difficulties when creating metamers. The solution found for this problem was to supply a greater voltage to the collector of the transistor. This was done by switching the source of the voltage from the Arduino which had a max output of 5 V to the battery pack on the PCB which had a max output of 7.5 V. When testing, it was typically closer to 6.5 V as measured by the onboard voltmeter.

The method used for taking data was effective but had two major obstacles: the efficacy of the diffuser, and the effective lower limit of output from the DAC. The chosen method to mix the light from the LEDs together was a mass of translucent beads contained in an Erlenmeyer flask. The problem with this method is the ratio of absorbance to the quality of the mix. In order to get sufficient light into the fiber optic cable, the thickness of the layer of beads between the cable and the light source needed to be thin such that the beads did not absorb too much light. In order for the light to properly mix, the thickness of the layer of beads between the cable and the light needed to be thick such that there was sufficient diffraction. The placement of the diffuser had to be tightly tolerated. To satisfy these bounds, the diffuser was held a few centimeters above the LEDs to allow for some light mixing before entering the diffuser. The LEDs' output light was at a relatively narrow angle as well. Although this did not drop the intensity of the light a great deal, some situational optimization of height took place with every data measurement. The positioning and angle of LEDs definitely affected the mix quality in different regions of the diffuser. Given that the LEDs have non-zero size and cannot be positioned to perfectly mix their light every time, this may have introduced some error to the data. The other obstacle was the lower voltage limit placed on the base of the transistor. This problem occurred because of the set voltage difference between the base and emitter of the transistor. In order for there to be 0.7 V between the base and emitter, at least 0.7 V needed to be supplied to the base (which was a DAC output of 575 out of 4095). While this was not the absolute minimum necessary to induce a current in the circuit, it was not possible to predict the current through the circuit without knowing this voltage drop. At low voltages supplied to the base of the transistor (~400 DAC output → ~0.49 V), a small, non-zero voltage could be expected and was found at the emitter end of the circuit. This means that the transistor did not always have exactly 0.7V across the emitter and the base, which may have introduced errors in both §§*Pre-Processing* and §§*Post-Processing*.

When processing our data, one major error occurred: impossible chromaticity values. For the two metamers, yellow and orange, the xy chromaticity values found from the measurements reported by the colorimeter, when transformed onto the xyY color-space, resulted in a point outside of the theoretical limit of human vision. This could have occurred due to the limited data from the AS7262. The AS7262 colorimeter only supplies data on 6 wavelengths of light. When that data is converted into cone response data, the result could output a combination of responses that would have been impossible given a continuous spectrum. It is also possible that a peak wavelength was outside the measured values, thereby making it impossible to create a Gaussian regression, throwing off the predicted cone responses. Additionally, the excitation purities of the LEDs were approximated to 1 for the transformations. Because the relative measured intensities of the LEDs were used with approximated purities, the resulting values from the transformation could lie outside the CIE 1931 xyY color space. Approximating the excitation purities to higher precisions may address this issue to a degree.

A possible improvement to this experiment might include using a better diffuser as one of the greatest contributors to data error may have been insufficiently mixed light. With a diffuser made out of material with a higher refractive index and lower absorption index, higher quality data collection may be possible. Also, the used setup was unstable; the amount of diffusive material, despite our best efforts, shifted somewhat during data collection. A more secure setup where these variables would be constant would have produced more precise data. Another possible improvement would be supplying power through a power supply. The best way to simplify data processing would be supplying a set amount of power to the LED rather than a voltage to the base of a transistor. This would have simplified the calculations to supply a specific amount of power to the LED as well as removing the error from the lower limit of the transistor base voltage. It also would have reduced dependency on approximations and analysis through hardware, which would have reduced error. Another way to resolve this would have been to actively measure the voltage on the emitter end of the transistor and ‘search’ for the correct DAC output rather than pre-calculating it. This would have accounted for variations in the voltage drop.

Conclusion

Going into this project, there were two goals. First – we were to use LEDs with wavelengths different from those used in standard monitors and screens in order to increase the gamut of possible colors displayed. Ideally but unrealistically, this would expand the displayed colors to cover the full range of human color vision. Expanding the display gamut while still covering currently displayable colors using three LEDs was found to be somewhat infeasible. This is likely because, if LEDs that allowed for this were available, they would already be used as a standard for creating monitors and screens, maintaining the problem for this project. To rectify this, more than three LEDs were used. Aside from red, green, and blue LEDs, yellow and near-ultraviolet LEDs were added. Despite this, finding metamerisms significantly outside of the classic rgb color space proved to be arduously difficult, although we did find some values for metamers.

The second goal was to study human color vision – specifically, metamerism – using the available LEDs. This goal was more successfully achieved; multiple metamer pairs were found that utilized different LEDs. All of these pairs were both qualitatively and quantitatively consistent in terms of cone response.

Lessons

By conducting this experiment, we learned that humans perceive color in a highly biased fashion; there is more detection capability in some regions compared to others. Two of the three cone receptors have a significant overlap in the range of wavelengths to which they are sensitive to. As a result of this, there are many different conventions associated with determining color. Working with these standards and converting between them was not an insignificant consideration. Finally, we learned that commonly used approximations about the behavior of transistors do not hold at the lowest voltages and currents that we used to tune the intensities. Determining the correct behavior of the LEDs, transistors, and potentiometers in our circuit was a great obstacle in our experiment.

Usefulness of Device

The device we created, while useful in studying human color vision, is not effective in expanding traditional color gamut to the extent that it would be viable to replace any current standards. Adding a fourth or fifth LED into each pixel would increase space requirements per pixel, making the quality of the image worse despite the color accuracy increasing in certain circumstances as many good displays rely on denser pixel density. Replacing one of the traditional red, green, or blue LEDs would likely reduce the gamut and make color reproduction more complex. Creating a screen with better color accuracy has significant economic incentive behind it, so it makes sense that the screens in the current market are optimized to begin with. Many professions already use color-accurate and highly-tuned displays to do professional color grading and editing for marketing and content purposes. There are also calibration devices to attach to displays for color accuracy already on the market. For these reasons, it seems inadvisable to use our device in a monitor for the purposes of selling them, however it may be worth creating them to send to researchers studying the human eye and how it responds to light.

Acknowledgements

We would like to thank Professor George Gollin and Ivan Velkovsky in providing the materials and guidance needed to design and test our project.

References

Figures

1. IT Media. (2008, November 11). *Understanding the Colour Gamut of an LCD Monitor*. https://www.eizo.be/en/knowledge/monitor-expertise/understanding-color-gamut/
2. Burghardt, J. (2013, August 19). *Distribution of Cones and Rods on Human Retina.png*. Wikimedia Commons. https://commons.wikimedia.org/w/index.php?curid=37671985
3. BenRG. (2009, September 22). *Cone-fundamentals-with-srgb-spectrum.svg*. Wikimedia Commons. https://commons.wikimedia.org/w/index.php?curid=7873848
4. cmglee, Vanessaezekowitz. (2007 October 26). *Metamerism spectrum example.svg*. Wikimedia Commons. https://commons.wikimedia.org/w/index.php?curid=94744983
5. Németh, László. (2013, June 25). *RGB color wheel 24.svg*. Wikimedia Commons. https://commons.wikimedia.org/w/index.php?curid=26860820
6. Algr. (2014, February 7). *Luma Chroma both.png*. Wikimedia Commons. https://commons.wikimedia.org/w/index.php?curid=31044071
7. PAR. (2005, June 21). *CIExy1931.png*. Wikimedia Commons. https://commons.wikimedia.org/w/index.php?curid=194805
8. Cpesacreta. (2007, May 17). *Colorspace.png*. Wikimedia Commons. https://commons.wikimedia.org/w/index.php?curid=8359333
9. XKCD. (n.d.). *Color Models*. xkcd.com. *https://imgs.xkcd.com/comics/color_models.png*
10. Picture taken by researchers
11. Picture taken by researchers
12. Picture taken by researchers
13. Siepert, Bryan. (2021, November 15). *Adafruit MCP4728 I2C Quad DAC*. Adafruit. https://cdn-learn.adafruit.com/downloads/pdf/adafruit-mcp4728-i2c-quad-dac.pdf
14. Picture taken by researchers
15. Adafruit Industries. (2017, March 17). *AS7262 6-Channel Visible Spectral_ID Device with Electronic Shutter and Smart Interface*. Retrieved December 2, 2021, from

https://cdn-learn.adafruit.com/assets/assets/000/052/623/original/AS7262_DS000486_2-00_%281%29.pdf?1522179774.

16. Picture taken by researchers
17. Picture taken by researchers
18. Picture taken by researchers
19. Picture taken by researchers
20. George Gollin
21. Picture taken by researchers
22. Color and Vision Research Laboratory. *Cone Fundamentals*. http://cvrl.ucl.ac.uk/cones.htm
23. Picture taken by researchers
24. Picture taken by researchers
25. Picture taken by researchers
26. PolBr. (2021, January 5). *SRGB chromaticity CIE1931.svg*. Wikimedia Commons. Edited by researchers. https://commons.wikimedia.org/w/index.php?curid=98545086
27. PolBr. (2021, January 5). *SRGB chromaticity CIE1931.svg*. Wikimedia Commons. Edited by researchers. https://commons.wikimedia.org/w/index.php?curid=98545086
28. PolBr. (2021, January 5). *SRGB chromaticity CIE1931.svg*. Wikimedia Commons. Edited by researchers. https://commons.wikimedia.org/w/index.php?curid=98545086

Works Cited

Adafruit Industries. (2017, March 17). *AS7262 6-Channel Visible Spectral_ID Device with Electronic Shutter and Smart Interface*. Retrieved December 2, 2021, from https://cdn-learn.adafruit.com/assets/assets/000/052/623/original/AS7262_DS000486_2-00_%281%29.pdf?1522179774.

Adafruit Industries. (n.d.). *Super Bright Blue 5mm LED (25 pack)*. Adafruit Industries Blog RSS. Retrieved December 2, 2021, from https://www.adafruit.com/product/301.

Adafruit Industries. (n.d.). *Super Bright Green 5mm LED (25 pack)*. Adafruit Industries Blog RSS. Retrieved December 2, 2021, from https://www.adafruit.com/product/300.

Adafruit Industries. (n.d.). *Super Bright Red 5mm LED (25 pack)*. Adafruit Industries Blog RSS. Retrieved December 2, 2021, from https://www.adafruit.com/product/297.

Adafruit Industries. (n.d.). *Super Bright Yellow 5mm LED (25 pack)*. Adafruit Industries Blog RSS. Retrieved December 2, 2021, from https://www.adafruit.com/product/2700.

Arduino. (n.d.). *Arduino Mega 2560 REV3*. Arduino Online Shop. Retrieved December 2, 2021, from https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us.

Clark, Roger. (2019, May 23). *Color Part 1: CIE Chromaticity and Perception*. ClarkVision. https://clarkvision.com/articles/color-cie-chromaticity-and-perception/

Color and Vision Research Laboratory. *Cone Fundamentals*. http://cvrl.ucl.ac.uk/cones.htm

Commission Internationale de l'Éclairage. (2020). *E-LIV*. https://cie.co.at/e-ilv

Appendix A: Definition of Terms

Words are defined both as used in this paper and with their official CIE definition, where applicable, from the *ILV: International Lighting Vocabulary, 2nd Edition* (CIE S 017/E:2020).

Achromatic - Not having any Hue; neutral.

Achromatic Color - A Color without Hue; a white, gray, or black, or a colorless or neutral emission.
CIE LIV 17-22-049: Perceived colour devoid of hue.

Brightness - A perception resulting from the luminance of something, generally increasing with luminance, but is not a direct linear relation of luminance. It is how much something appears to emit light, similar to Lightness.
CIE LIV 17-22-059: Attribute of a visual perception according to which an area appears to emit, transmit or reflect, more or less light.

Chroma - A color's perceived difference from a grey of the same lightness; this compares a color to a gray of the same brightness. A measure of chromatic intensity, similar to Colorfulness, Excitation Purity, and Saturation.
CIE LIV 17-22-074: Colourfulness of an area judged as a proportion of the brightness of a similarly illuminated area that appears grey, white or highly transmitting.

Chrominance - The Color information independent of Luminance information, especially for images or video; essentially a set of one or more Chromaticity values.

Chromaticity - Properties defining a color regardless of Luminance, usually Hue and Colorfulness.
CIE LIV 17-23-052: Property of a colour stimulus defined by its chromaticity coordinates, or by its dominant or complementary wavelength and purity taken together

Chromaticity Diagram - A visual representation of chromatic properties for a given color model. A common example is the CIE 1931 xy chromaticity diagram (see Figure 7).
CIE LIV 17-23-054: Plane diagram in which points specified by Chromaticity coordinates represent the chromaticities of colour stimuli.

Color - A given perception of color vision, usually categorized into groups of various hues and tints/shades: reds, greens, grays, et cetera. In humans this is a result of a system of Cones and processes which distinguishes light based on wavelengths and intensities. Colors can be defined within a Color Space using numerical values, usually Hue, Saturation, and Relative Luminance. See Colour.

Color Matching Functions - Mathematical description of the chromatic response of an observer, usually spectral sensitivity curves corresponding to CIE tristimulus values. See Trichromatic System.

Color Model - A mathematical representation of colors by a tuple of values, typically three or four. Examples would be an RGB color model with primaries of red, green and blue or the CMYK model for dyes and inks on a substrate. See Colour Appearance Model.

Color Space - A specific organization of colors for the reproduction of color representations. This can be a set defined by pigments/swatches or mathematically determined. A color space defines a particular combination of a Color Model and mapping functions. This is often informally used to identify a Color Model, but this is not strictly correct usage: Though a given Color Space does identify a specific Color Model, the given Color Space has specific mappings for the Color Model. A Color Space links the Color Model and the defined Color Matching Functions. See Colorimetric Colour Space.

Color Wheel - A cyclic representation of color Hues. See Figure 5.

Colorfulness - The intensity of a color, usually dependent on reflectance and illumination, and generally increases with brightness. A measure of chromatic intensity, similar to Chroma, Excitation Purity, and Saturation.
CIE LIV 17-22-072: Attribute of a visual perception according to which the perceived colour of an area appears to be more or less chromatic.

Colorimetric Colour Space - The CIE defined term corresponding to Color Space.
CIE LIV 17-23-042: Colour space defined by three colorimetric coordinates.

Colour - The CIE defined term corresponding to Color.
CIE LIV 17-22-040: A characteristic of visual perception that can be described by attributes of hue, brightness (or lightness), and colourfulness (or saturation or chroma).

Colour Gamut - The CIE defined term corresponding to Gamut.
CIE LIV 17-32-007: Volume, area, or solid in a colour space, consisting of all those colours that are either (a) present in a specific scene, artwork, photograph, photomechanical, or other reproduction; or (b) capable of being created using a particular output device and/or medium.

Colour Appearance Model - The CIE defined term corresponding to Color Model.

CIE LIV 17-23-027: Model describing colour appearance, built from descriptors of colour stimuli combined with the illuminating and viewing environment.

Cone - A cell in the human eye specialized to a range of wavelengths. By taking the responsivity of multiple cones sensitive to different ranges, the brain is able to discriminate between most spectra. These cells are located throughout the retina, but have a higher concentration in the fovea, at the center of vision.

CIE LIV 17-22-002: Photoreceptors in the retina containing light-sensitive pigments capable of initiating the process of photopic vision.

Excitation Purity - The ratio of the distances from the defined white point to a given color and from the white point to the Spectrum Locus or the Purple Boundary. This is calculated from the xy coordinates of a Chromaticity Diagram. A measure of chromatic intensity, similar to Chroma, Colorfulness, and Saturation.

CIE LIV 17-23-066: Quotient NC/ND of two collinear distances on the chromaticity diagram of the CIE 1931 or 1964 standard colorimetric systems, the first distance being that between the point C representing the colour stimulus considered and the point N representing the specified achromatic stimulus, the second distance being that between the point N and the point D on the spectrum locus at the dominant wavelength of the colour stimulus considered, leading to the following expressions: $p_e = \frac{y - y_n}{y_d - y_n}$ and $p_e = \frac{x - x_n}{x_d - x_n}$, where (x, y), (x_n, y_n), (x_d, y_d) are the x, y chromaticity coordinates of the points C, N, and D respectively.

Fluorescence - Emission of light from a substance which previously absorbed light or electromagnetic radiation. The emitted light is often a lower frequency than the absorbed radiation. This causes the common occurrence of vibrant "Fluorescent" or "Highlighter" colors due to an absorption of invisible UV radiation and reemission in the visual spectrum, resulting in a color that appears more vivid than would be expected in some given lighting conditions.

CIE LIV 17-24-023: Emission of optical radiation when a substance is exposed to any type of electromagnetic radiation, where the emitted radiation generally appears within 10 ns after the excitation.

Full Width at Half Maximum (FWHM) - The range between the two values at which a signal is at half of its maximum.

Gamut - The subset of colors that are accurately capturable or displayable by or on a certain device, or the colors that are present in a representation. When defined using a given color space, this is a shape determined by the sensors or emitters of a device or a definite set of values within that space. For a typical monitor defined by a chromatic color space diagram (such as CIE1931xy), this is often a triangle which has vertices at the chromaticity values of the subpixel emitters. A gamut is often compared to a defined color space, such as sRGB. Overlap of the gamut and color space shows the accurate colors within that space that a device with that gamut will accurately display or capture. See Colour Gamut.

Hue - The degree to which a color appears to be red, orange, yellow, green, blue, and violet or magenta. This quality is independent of Brightness and Colorfulness. This is often represented by a Color Wheel, the dominant wavelength of light, or the wavelength of complementary color of light. It should be noted that under this definition, the color "brown" is of an orange hue with a low lightness and/or low saturation.

CIE LIV 17-22-067: Attribute of a visual perception according to which an area appears to be similar to one of the colours red, yellow, green, and blue, or to a combination of adjacent pairs of these colours considered in a closed ring.

Integrated Development Environment (IDE) - A piece of software that contains tools that a computer programmer would use when writing and running code. Many of these tools assist in debugging code.

Lightness - A perception of how bright an object is when compared to a similarly lit white. This is similar to Brightness, but is a comparison.

CIE LIV 17-22-063: Brightness of an area judged relative to the brightness of a similarly illuminated area that appears to be white or highly transmitting.

Luminance - A measure of the perceived brightness of light, with SI units of candela per square meter or, more commonly referred to when comparing display technologies, nits; this is different from Brightness and Lightness which are the perceptions of said luminance.

CIE LIV 17-21-050: Density of luminous intensity with respect to projected area in a specified direction at a specified point on a real or imaginary surface $L_v = \frac{dI_v}{dA}\frac{1}{cos(\alpha)}$ where I_v is luminous intensity, A is area and α is the angle between the normal to the surface at the specified point and the specified direction.

Luminance, Relative (Y) - While strict luminance is measured in SI units, Relative Luminance is normalized from 0.0 to 1.0 (1 or 100). The high value of 1.0 (or 100) indicates perfect reflection or emission of a reference white. This is usually designated as "Y" (capitalized) for the xyY or XYZ color spaces.

Metamers - Colors that can be constructed from two or more spectral distributions; Colors that reflect Metamerism.

CIE LIV 17-23-008: Spectrally different colour stimuli that have the same tristimulus values in a specified colorimetric system.

Metamerism - The phenomenon of the same perceived color arising from different spectral distributions.
CIE LIV 17-23-006: Property of spectrally different colour stimuli that have the same tristimulus values in a specified colorimetric system.

Monochromatic - Consisting of only one color, or of only one or of only an extremely narrow band of wavelengths of light. See Monochromatic Radiation.

Monochromatic Radiation - Consisting of only one or of only an extremely narrow band of wavelengths of light.
CIE LIV 17-21-014: Radiation characterized by a single frequency or a single wavelength.

Photopic Vision - Vision of the eye under well-lit conditions allowing for color vision due to the use of cone cells. This usually has much better visual acuity than Scotopic Vision.
CIE LIV 17-22-016: Vision by the normal eye in which cones are the principal active photoreceptors.

Purple Boundary - The straight outer edge of a Chromaticity Diagram connecting the long and short wavelength ends of the Spectrum Locus, represented by a continuum of magentas.
CIE LIV 17-23-058: Line in a chromaticity diagram, or the plane surface in a tristimulus space, that represents additive mixtures of monochromatic stimuli of wavelengths approximately 380 nm and 780 nm.

Qualia - Subjective, conscious experience; the event as perceived by the mind.

Rod - A small photoreceptor cell in the human eye capable of sensing a quantity of light. Incapable of distinguishing color, but more sensitive than cones to low-light conditions. Mostly situated outside the fovea and the center of vision.
CIE LIV 17-22-003: Photoreceptors in the retina containing a light-sensitive pigment capable of initiating the process of scotopic vision.

Saturation - A color's perceived difference from white; this compares a color to itself. A measure of chromatic intensity, similar to Chroma, Colorfulness, and Excitation Purity.
CIE LIV 17-22-073: Colourfulness of an area judged in proportion to its brightness.

Scotopic Vision - Vision of the eye under low-light conditions through the use of rod cells. Often characterized by a lack of color perception and a shift of visual sensitivity to shorter wavelengths. This usually has much worse visual acuity than Photopic Vision.
CIE LIV 17-22-017: Vision by the normal eye in which rods are the principal active photoreceptors.

Serial Monitor - A tool in the Arduino IDE that allows a programmer to interact with the Arduino. Programmers can send text commands to the serial monitor or program the Arduino to output text to the serial monitor.

Spectrum - (pl. Spectra) A narrow wavelength or a range of wavelengths of light.
CIE LIV 17-21-015: Display or specification of the monochromatic components of the radiation considered.

Spectrum Locus - The curved line creating the outer edge of a Chromaticity Diagram which represents the pure monochromatic emission of a Hue. This is sometimes labeled with the wavelengths corresponding to various pure emissions.
CIE LIV 17-23-056: Locus, in a chromaticity diagram or in a tristimulus space, of points that represent monochromatic stimuli.

sRGB - A widely used Color Space standardized with three color primaries of specific values, one each of red, green, and blue. It is commonly used for reference and comparison.
CIE LIV 17-32-065: Colour space defined by the IEC.

Trichromatic System - The CIE defined term corresponding to Color Matching Functions.
CIE LIV 17-23-036: System for specifying colour stimuli in terms of tristimulus values, based on matching colours by additive mixture of three suitably chosen reference colour stimuli.

Tristimulus Values - A set of three values that correspond to amounts of three primary stimuli or colors required to represent the stimulus of a given color. In some Color Spaces, such as the LMS and XYZ spaces, these primaries are not directly real colors. For example, the CIE 1931 XYZ Color Space has its values defined as “X,” “Y,” and “Z,” with Y describing perceived Brightness and X & Z being “imaginary” primaries that all combine to determine real color values. (1,0,0), (0,1,0), and (0,0,1) are all “imaginary” colors which lie outside the real colors.
CIE LIV 17-23-038: Amounts of the reference colour stimuli, in a given trichromatic system, required to match the colour of the stimulus considered.