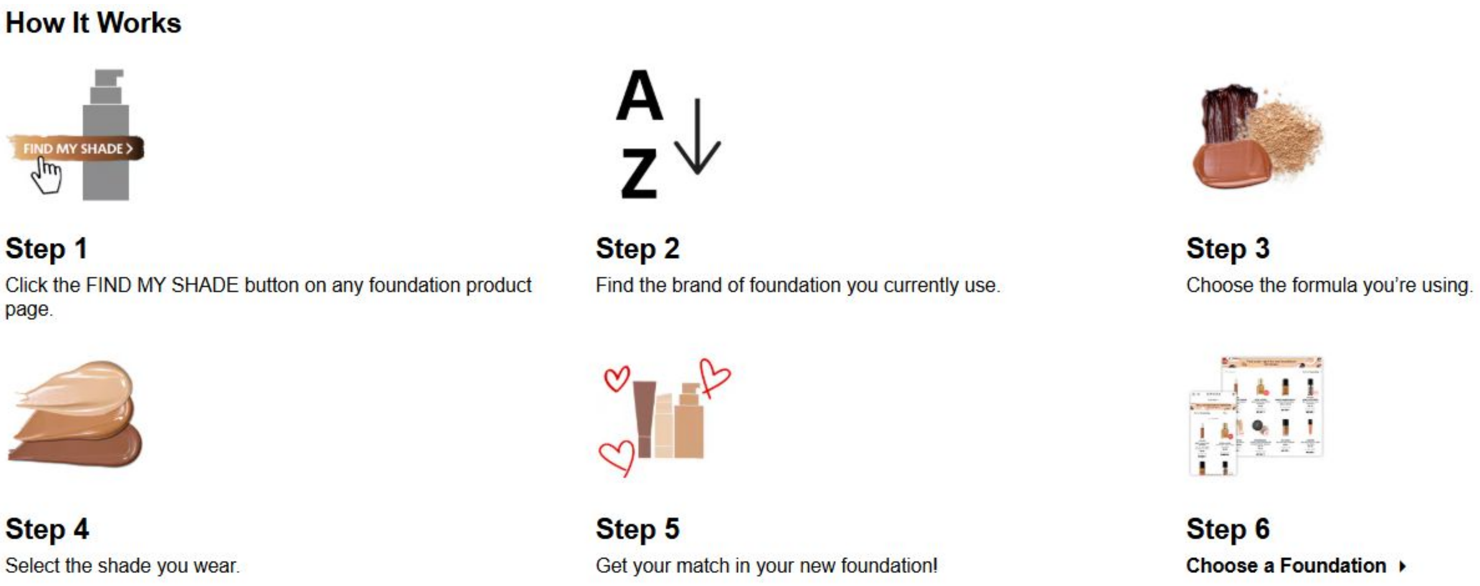
Colorimeter for Skin Tone & Makeup Applications

Electrical & Computer Engineering

Group 6: Ashley Herce, Waidat Bada, Shriya Surti

Problem

- Skin tone matching in beauty products is often inaccurate due to lighting



Problem

- Mismatched products lead to high return rates and waste as many retailers have generous return policies

Returns Policy

If you are not completely satisfied with a Sephora purchase or gift for any reason, Sephora welcomes you to return new or gently used products for a full refund to your original method of payment if returned within 30 days of purchase, in most cases. In order to complete the return, you must have a proof of purchase.

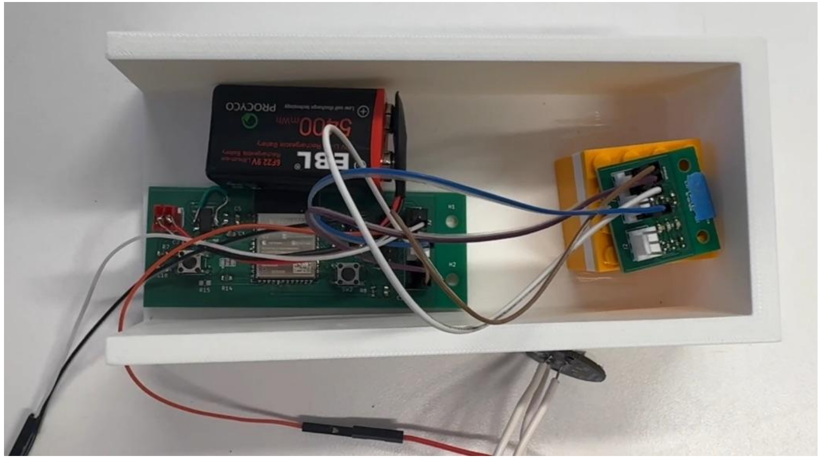
Please review our return, refund, and exchange policies below for purchases made on Sephora.com, purchases made in Sephora stores, and purchases made on third party on-demand delivery marketplaces such as Instacart, Doordash and Shipt. Products must be returned in new or gently used condition. Sephora monitors return activity for abuse and reserves the right to limit returns or exchanges at Sephora in all instances. Beauty Insider points and spend associated with all refunds will be removed when the transaction is processed. All returns are subject to validation and approval at Sephora's discretion. If a return is not approved by Sephora for any reason, the item may not be returned to you. We may ask you for a driver's license or government ID to verify your identity.

Problem

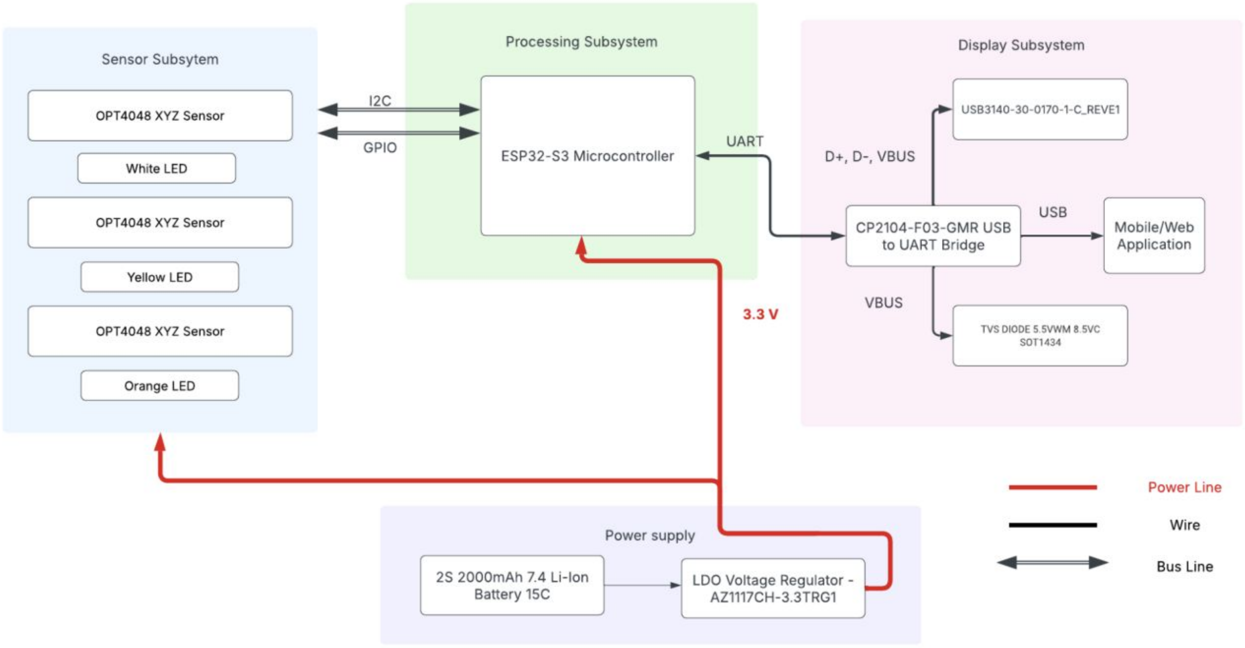
- Consumers with complex undertones face frustration and unmet needs

Our Solution

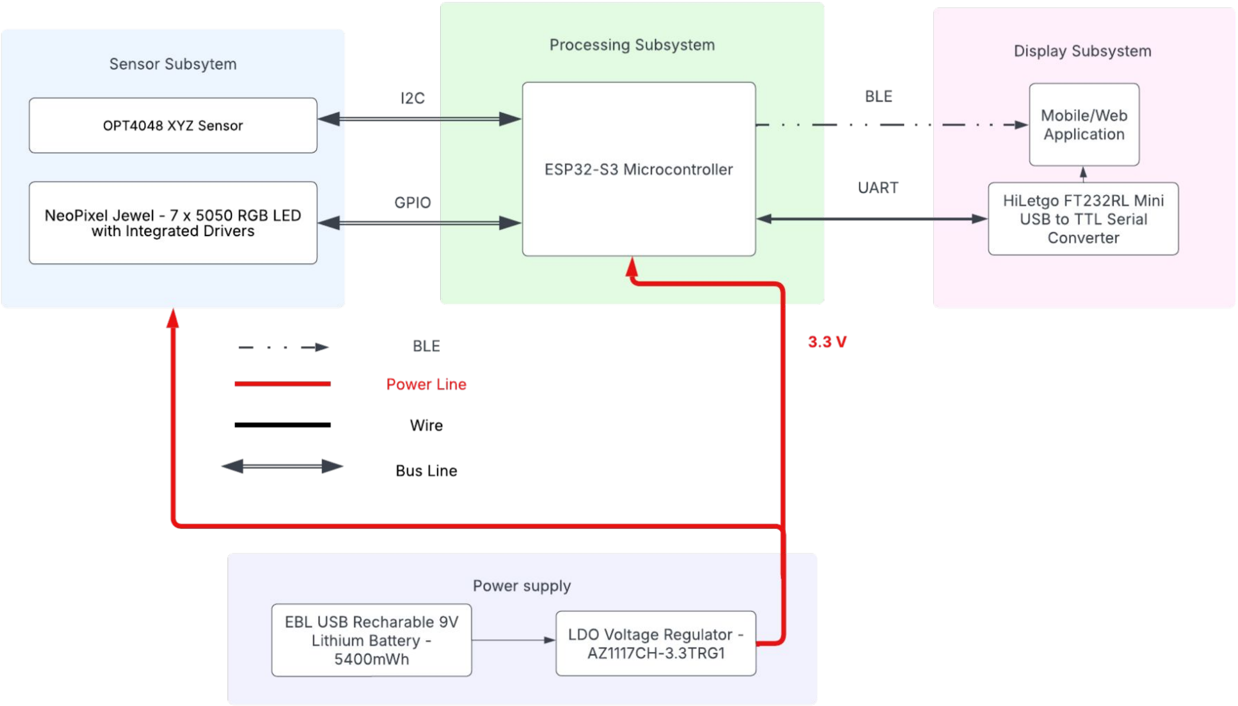
- A web-connected colorimeter uses backlights and XYZ sensors to capture accurate skin tone across lighting conditions
- Data is processed and sent to a web app for personalized, product recommendations
- Utilizes a more inclusive skin tone scale to ensure better representation of all complexions



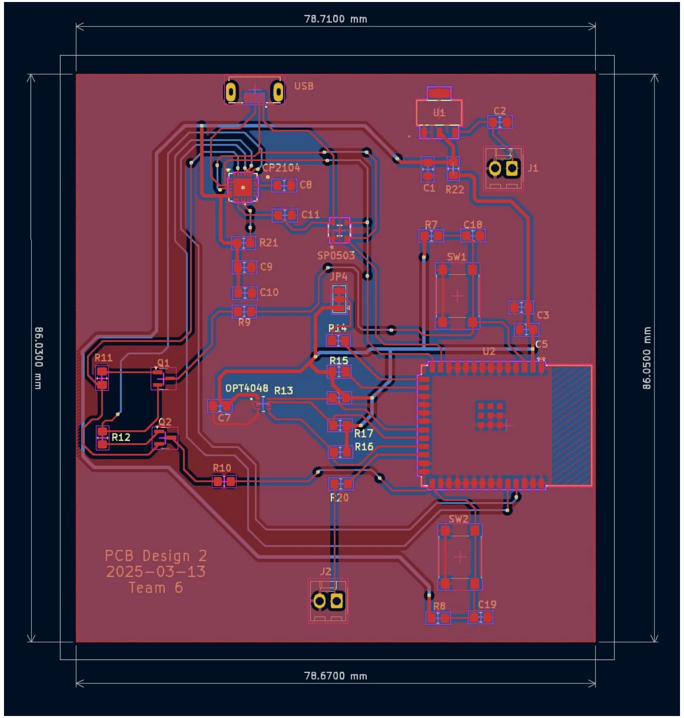
Original Design

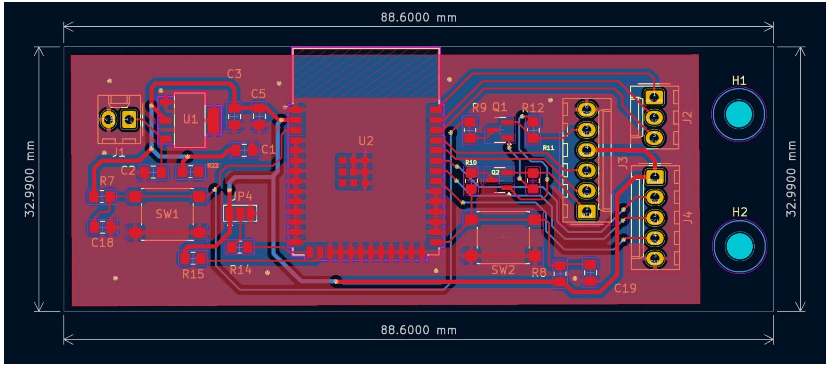


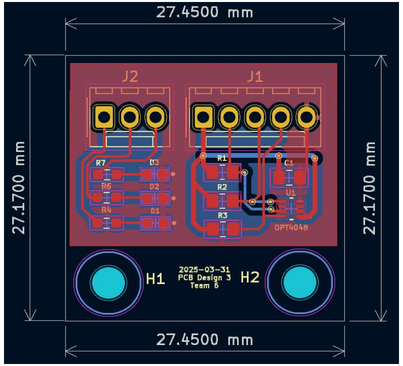
Final Design

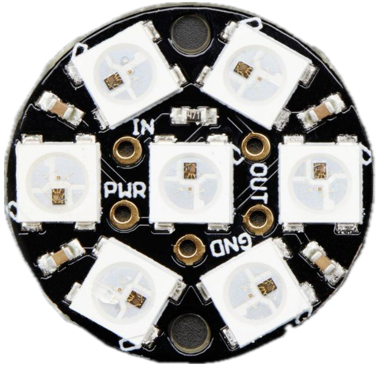


PCB Design Changes









Power Subsystem

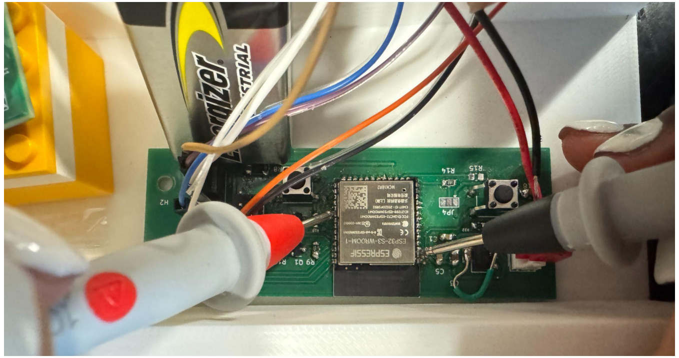
Power Subsystem

Requirements

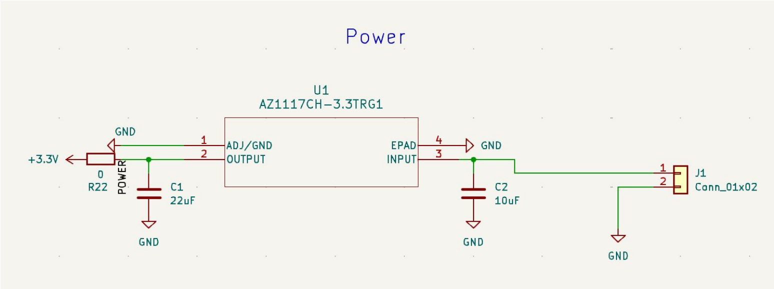
- Provide 30 A average/continuous current
- Provide stable 3.3V ± 5%

Verifications

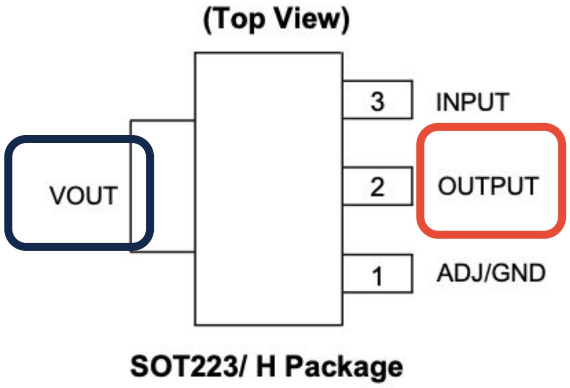
- Use a low-resistance shunt resistor to measure voltage drop
- Use a multimeter to measure the voltage drop (V) across the resistor
- Use the multimeter for continuity testing
- Use an oscilloscope to measure the output voltage ripple under various load conditions

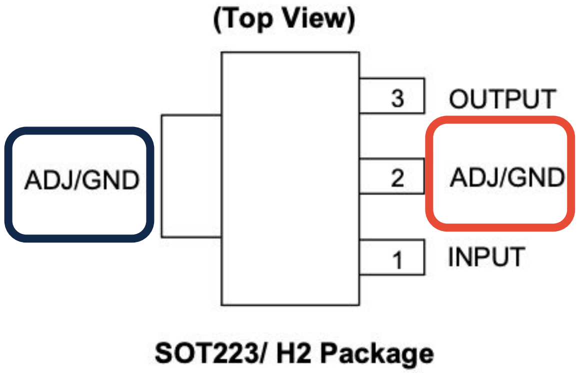


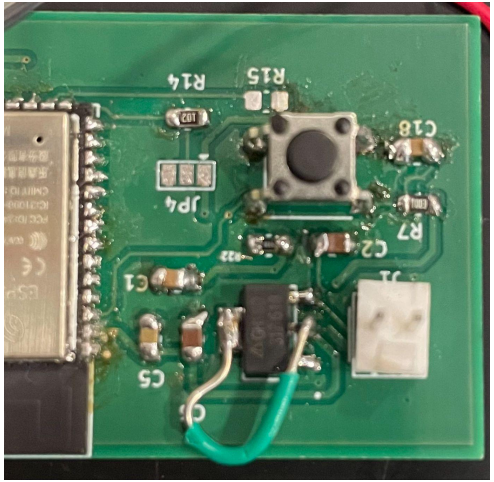
Power Subsystem



- Encountered issues with our LDO regulator, as it caused a short in our circuit when supplying the PCB with 9V
- This prompted us to investigate both the component footprint and its datasheet



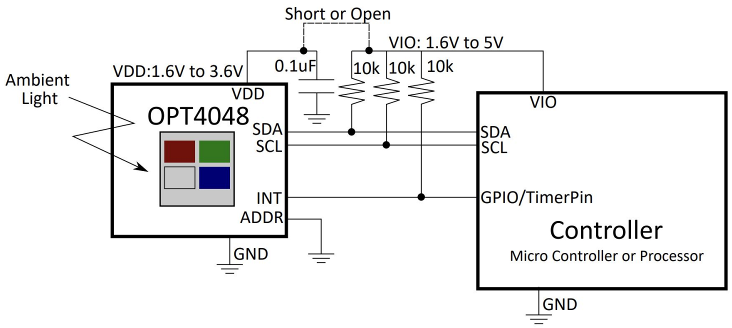




Sensor Subsystem

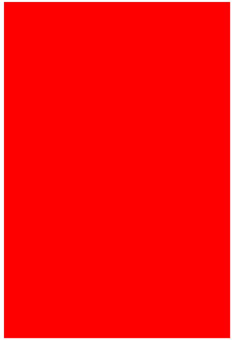
Requirements

- The OPT4048 color sensor must detect XYZ values with an accuracy of 75% compared to any/all standardized input colors
- The color detection latency must be less than 150 ms after the initial set up of I2C communication through the ESP32 microcontroller
- The LED must be able be able to switch ambient lighting conditions with a latency of less than or equal to 200 ms to mimic different environmental conditions

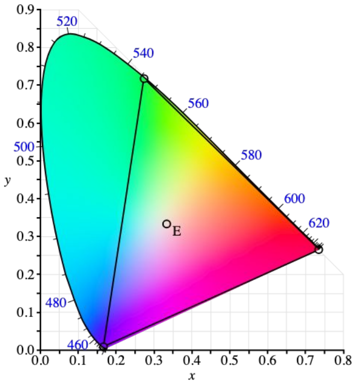


Verifications

- Capture and record the XYZ values detected by the sensor.
- Compare the detected values with the reference values by analyzing the individual X, Y, and Z components and calculating the differences between the experimental and actual values for each to determine if the accuracy meets or exceeds 85%.

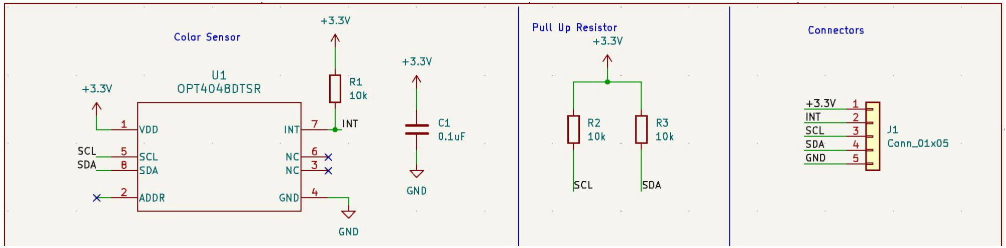


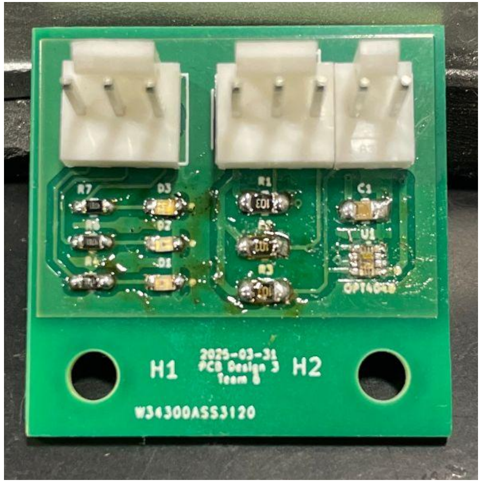
```
09:33:31.842 ->  CIEy: 0.34
09:33:32.038 -> CIEx: 0.63
09:33:32.038 ->  CIEy: 0.34
09:33:32.234 -> CIEx: 0.63
09:33:32.234 ->  CIEy: 0.34
09:33:32.463 -> CIEx: 0.63
09:33:32.463 ->  CIEy: 0.34
09:33:32.660 -> CIEx: 0.63
09:33:32.660 ->  CIEy: 0.34
09:33:32.858 -> CIEx: 0.63
09:33:32.858 ->  CIEy: 0.34
09:33:33.056 -> CIEx: 0.63
09:33:33.056 ->  CIEy: 0.34
```



Color Sensor Setup

- **Conversion Time:** higher measurement resolutions and noise reduction with longer conversions (set to 200ms)
- **Operation Mode:** One Shot to ensure sensor only measures when triggered to minimize thermal noise and power consumption





Note

- OPT4048 marketed as a object color sensor but is only effective for screens & backlit items not skin

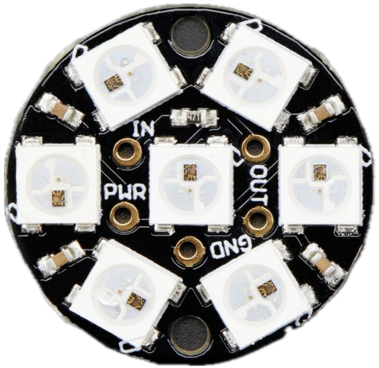
Requirements

- Display realistic ambient conditions
- LEDs must work concurrently with OPT4048 w/o I2C/SPI conflicts
- Neopixel must work along with the BLE and without introducing timing issues

Verification

- Visual inspection under controlled conditions to confirm lighting matches expected tone
- Connect both devices and run test scripts that trigger simultaneous read/write operations
- Ensure repeated, accurate readings from the color sensor while the Neopixel is active
- Run BLE communication routines while updating Neopixel output

Light Sensor




```
struct LightingCondition {
  uint8_t r, g, b;
} lightingConditions[3] = {
    {64, 156, 255},   // Daylight
    {255, 255, 251},  // Fluorescent
    {255, 147, 41}    // Incandescent
};
```

Results

- Encountered timing issues with the standard Adafruit Neopixel library
- FASTLED enabled stable operation alongside both the color sensor and BLE module
- Successfully replicated ambient lighting conditions through visual testing and iterative tuning

Modifications

Color Sensor

- Tested Monk 10 Color values on an iPad (Full Brightness) rather than paint samples and skin
    - Improved accuracy in color detection
- Captured XYZ values from ADC Channels and converted to xyY for higher accurate readings

Lighting Requirements

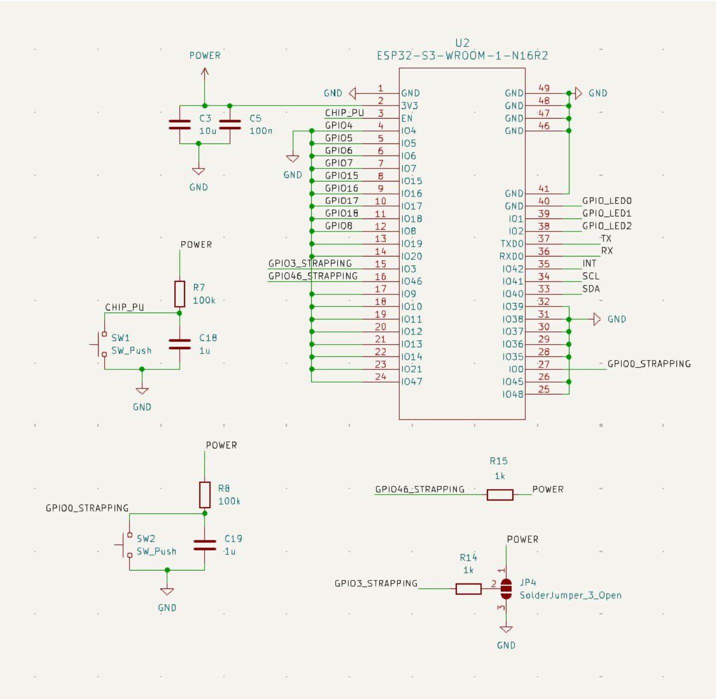
- PCB Mounted Lights did not provided effective lighting results
- Utilized single 5050 RGB LED mounted on Adafruit NeoPixel Jewel to mimic 3 lighting conditions at 100% brightness
- 5050 RGB LED has ~18mA constant current drive for consistent lighting

Processing Subsystem

Processing Subsystem

Requirements

- Support I2C and efficiently acquire sensor data
- Control RGB LED brightness and color settings
- Process raw color sensor data
- Convert raw data into HEX values to determine skin tone
- Establish stable Bluetooth connection for real-time data transmission



Verifications

- Connect voltmeter to SCL/SDA trigger a sensor read operation and measure the power to sensor
- Compare the processed HEX values with a reference color chart to validate accuracy
- Measure latency and packet loss using debugging tools such as ESP-IDF Bluetooth logs

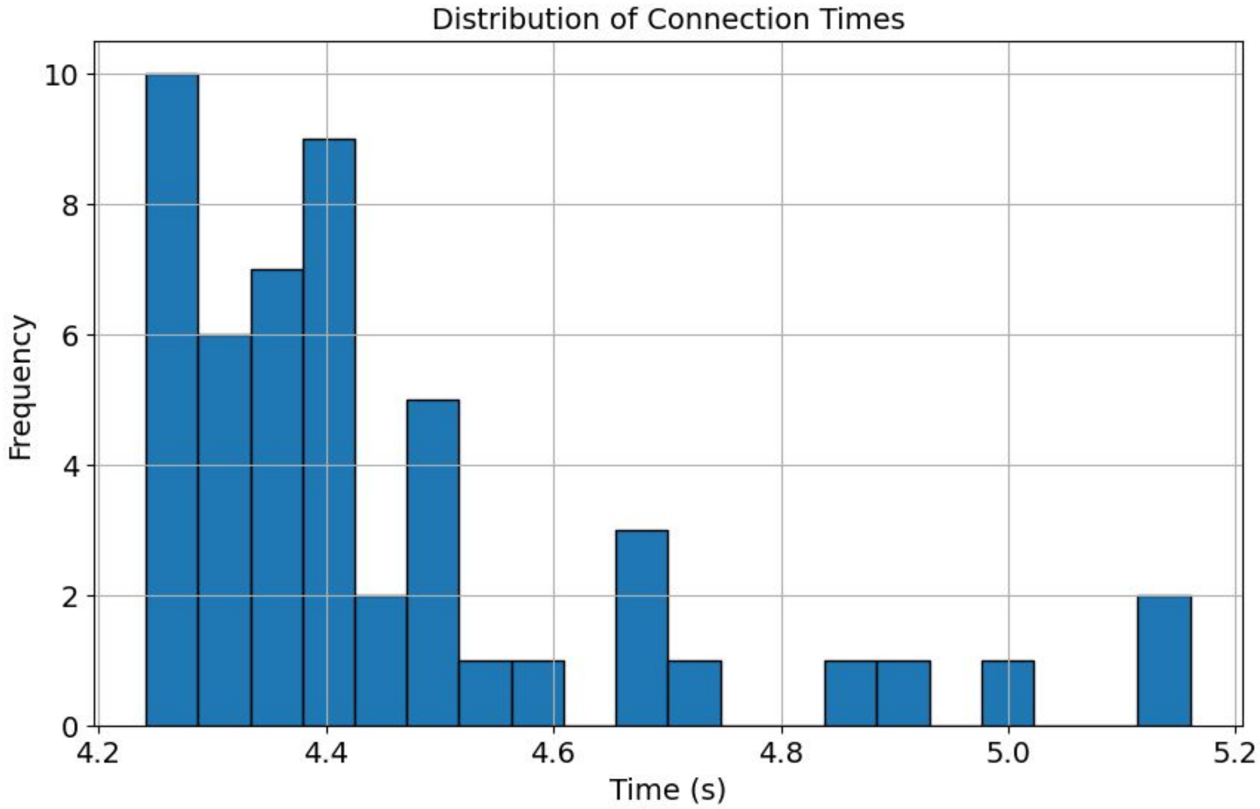
Results

- Utilized GPIOs to power Neopixel Jewel
  - Automatic brightness/color control
- Polled color sensor results
  - Verified real-time raw data capture
- Establish stable and continuous Bluetooth connection

```
2025-05-05 09:54:45,454 [INFO] Results for connection time:
2025-05-05 09:54:45,456 [INFO] Average delay: 4.46 seconds
2025-05-05 09:54:45,456 [INFO] Minimum delay: 4.24 seconds
2025-05-05 09:54:45,456 [INFO] Maximum delay: 5.16 seconds
```

```
PS C:\Users\Class\Desktop\colorMatch\server\Helper_Functions> python .\bluetooth_test.py
[INFO] SimpleBLE: D:\a\simpleble\simpleble\simpleble\src\backends\windows\Utils.cpp:33 in initialize_winrt: CoGetApartmentType: cotype=-1, qualifier=0, result=800401F0
[INFO] SimpleBLE: D:\a\simpleble\simpleble\simpleble\src\backends\windows\Utils.cpp:41 in initialize_winrt: RoInitialize: result=0
2025-05-05 09:36:51,236 [INFO] Found 1 adapter: DESKTOP - Front [18:cc:18:83:b5:cf]
2025-05-05 09:36:51,236 [INFO] Selected adapter: DESKTOP - Front [18:cc:18:83:b5:cf]
2025-05-05 09:36:53,328 [INFO] Connecting to: ESP32-team6 [74:4d:bd:20:a0:29]
2025-05-05 09:36:55,711 [INFO] Successfully connected
2025-05-05 09:36:55,711 [INFO] Time taken: 4.53 seconds
Press Enter to run connection test...
```

BLE Connection Times



Software Subsystem

Software Subsystem

Requirements

- Display real-time CIELAB values on the screen
- Implement tolerance calculations for perceptually accurate product matching
- Display product information that matches CIELAB values from the color sensor
- Classify the user's skin tone into Monk and Fitzpatrick Scale
- Ensure color sensors and database are fair and unbiased for multiple skin tones

Verifications

- Use a dataset of Lab values to calculate Delta E* values between predefined samples (target) and standards (shade values) and Delta E* meets threshold < 1.75 for acceptable matches
- Perform a query using sensor output and confirm correct product information is displayed
- Test communication between ESP32 and the database via BLE by sending queries and verifying responses using python scripts
- Use Delta E and confirm that shades are categorized based on their closest Lab value(s) in the database
- Compare the accuracy metrics across different skin tone categories. Significant differences in accuracy will indicate potential bias

Background Information: XYZ vs xyY Color Space

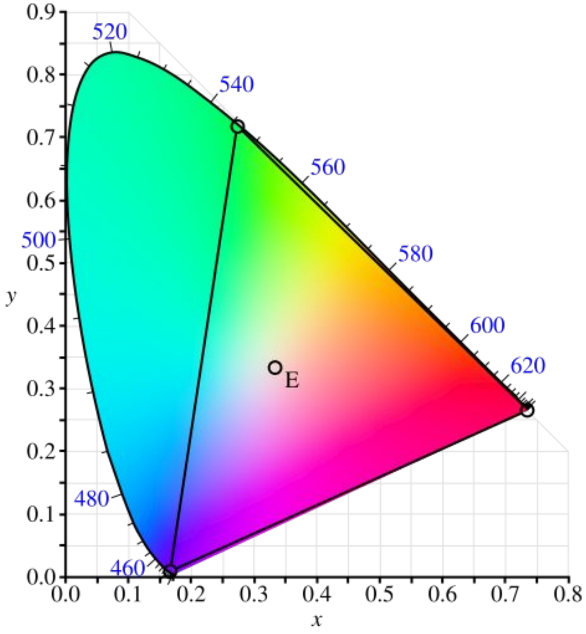
XYZ

- Tristimulus Color Space
- X: Red Green Axis
- Y: Luminance (Brightness)
- Z: Blue Axis
- Device Independent & Based on Human Vision
- All visible colors can be represented as positive values of X,Y,Z
- **Best for full spectral data & computation**
  (converting to/from other spaces)

xyY

- Chromaticity Color Space derived from XYZ
  - Defines color hue & saturation
- Separates color (chromaticity) from brightness (Luminance)
- Projects color int 2D (x,y) color space with a separate brightness (Y)
- **Best for perceptual color accuracy, comparison or visualization**

$$CIEx = \frac{x}{x+y+z} \qquad CIEy = \frac{y}{x+y+z}$$



Software Video Overview

Database Design & Product Distribution

Fitzpatrick	Category	Monk	# Of Products
I	Very Fair	1	33
		2	215
II	Fair	3	114
		4	751
III	Medium	5	3649
		6	2813
IV	Olive	7	1333
		8	796
V	Brown	9	103
		10	39
VI	Dark Brown		

- Product Shades categorized into 1 of 10 Monk Categories based on < ΔE
- Database locally hosted using SQLite and accessed via Python
  - 105 Brands
  - 478 Products
  - 9846 Shades

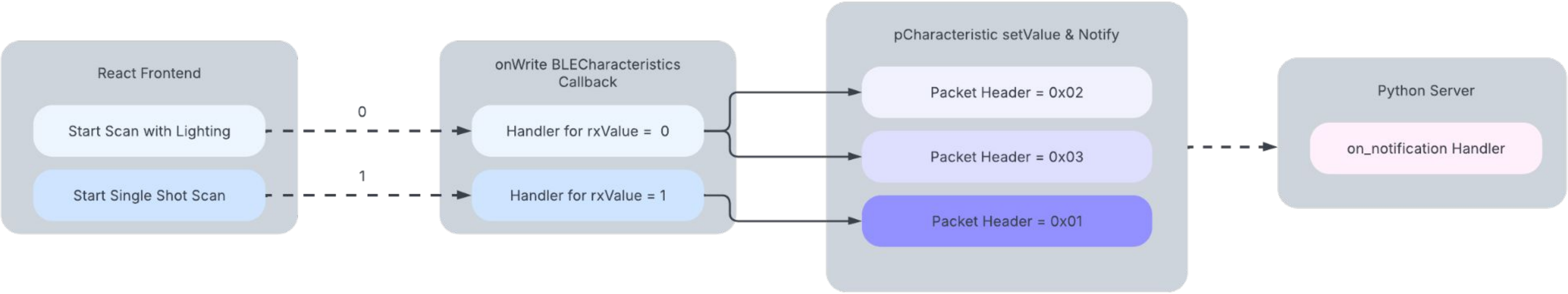
Threshold Requirements

- ΔE quantifies the difference between two colors in CIE L*a*b* color space (measures perceived color difference)
- $\Delta E \leq 3$ No significant difference an average person could perceive

$$\Delta E\ Threshold:\ \leq\ 1.75\ \ Percentage\ Format:\ \frac{100 - \Delta E}{100} \times 100 \approx 100\% - 98.35\%$$

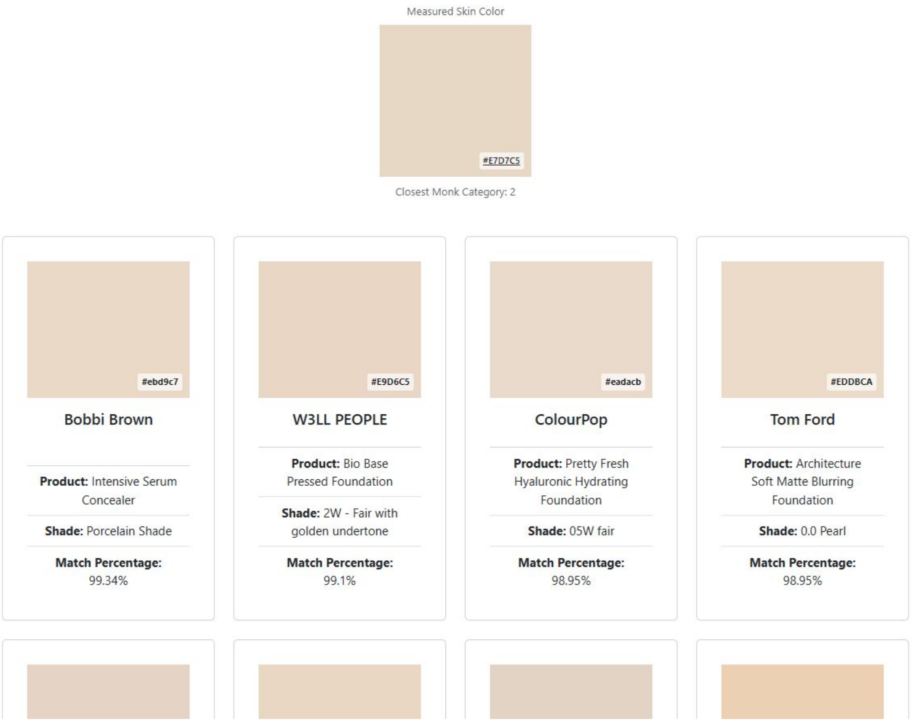
Major Components

- Uses Simple BLE and Python to connect to ESP32
- User Friendly React Based Application on Localhost
- **During Scanning**: Disable button to ensure user cannot send another request until first one is fulfilled



Displaying Products

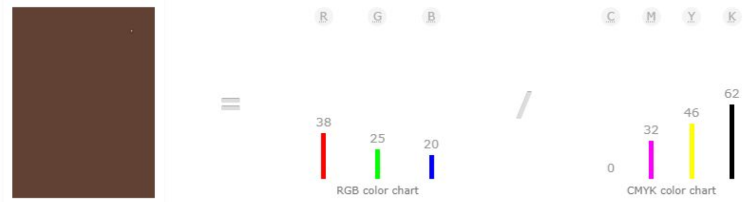
- Emit Products and captured data via Socket.IO
- **Socket.IO:** Allows communication via client and server (Python) without excessive polling
- Display Brand, Product, Shade, ΔE (as Percentage), Product Hex Value
- Display measured value (Hex Format) and Calculated Monk Category



Sensor Accuracy Results

Color Space Conversion (with Lighting)

Expected (Monk 8)



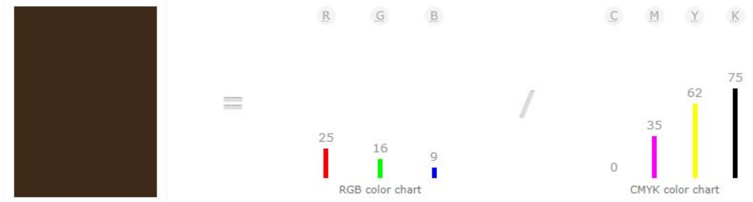
• #604134 color description : **Very dark desaturated orange.**

#604134 Color Conversion

The hexadecimal color #604134 has RGB values of R:96, G:65, B:52 and CMYK values of C:0, M:0.32, Y:0.46, K:0.62. Its decimal value is 6308148.

Hex triplet	604134	#604134	**CIE-LAB**	30.678, 11.667, 13.335
RGB Decimal	96, 65, 52	rgb(96,65,52)	**XYZ**	7.334, 6.516, 4.12
RGB Percent	37.6, 25.5, 20.4	rgb(37.6%,25.5%,20.…	**xyY**	0.408, 0.363, 6.516
CMYK	0, 32, 46, 62		**CIE-LCH**	30.678, 17.719, 48.816
HSL	17.7°, 29.7, 29	hsl(17.7,29.7%,29%)	**CIE-LUV**	30.678, 20.73, 12.378
HSV (or HSB)	17.7°, 45.8, 37.6		**Hunter-Lab**	25.526, 6.616, 8.298
Web Safe	663333	#663333	**Binary**	01100000, 01000001, 00110100

Measured (Monk 8)



• #3f2918 color description : **Very dark orange [Brown tone].**

#3f2918 Color Conversion

The hexadecimal color #3f2918 has RGB values of R:63, G:41, B:24 and CMYK values of C:0, M:0.35, Y:0.62, K:0.75. Its decimal value is 4139288.

Hex triplet	3f2918	#3f2918	**CIE-LAB**	18.838, 7.989, 15.207
RGB Decimal	63, 41, 24	rgb(63,41,24)	**XYZ**	3.008, 2.709, 1.229
RGB Percent	24.7, 16.1, 9.4	rgb(24.7%,16.1%,9.4…	**xyY**	0.433, 0.39, 2.709
CMYK	0, 35, 62, 75		**CIE-LCH**	18.838, 17.178, 62.286
HSL	26.2°, 44.8, 17.1	hsl(26.2,44.8%,17.1…	**CIE-LUV**	18.838, 13.807, 11.462
HSV (or HSB)	26.2°, 61.9, 24.7		**Hunter-Lab**	16.458, 3.819, 7.095
Web Safe	333300	#333300	**Binary**	00111111, 00101001, 00011000

Unpacking Data

Receive Binary xyY Message → Convert to XYZ → Convert to CIE L*a*b* → Categorize input to Monk Category → Query Database for Matching product within threshold

Calculated Average Results

Fluorescent Results

Daylight Results

Incandescent Results

Header 0x02 (Averaged xyY)

- *Measured*: [0.43, 0.39, **2.72**]
- *Expected*: [0.433, 0.39, **2.709**]

Header 0x03

- *Daylight*: [0.40, 0.37, 2.52]
- *Fluorescent*: [0.42, 0.38, 2.83]
- *Incandescent* [0.46, 0.40, 2.82]

Error Analysis (Average vs Reference)

- *Daylight*: [0.03, 0.01, 0.03]
- *Fluorescent*: [0.02,0.01, 0.01]
- *Incandescent* [0.20, 0.11, 0.10]

Conclusion

- Brighter lumiance under daylight
- Average most similar to fluorescent reference across all channels

Unpacking Data

Receive Binary xyY Message → Convert to XYZ → Convert to CIE L*a*b* → Categorize input to Monk Category → Query Database for Matching product within threshold

Expected Result

Calculated Average Results

Expected XYZ

- XYZ [3.0008, 2.709, 1.229]

Calculated XYZ

- XYZ [3.02, 2.72, 1.32]

```
2025-05-02 10:33:22,856 [INFO] Sent: 0
Received average xyY: [0.4275, 0.3854, 2.7239]
Recieved xyY values: [0.43, 0.39, 2.72]
XYZ Values Converted: [3.02, 2.72, 1.32]
Converted LAB Values: [18.90, 7.95, 14.20]
Monk Category: 9
```

$$X = \frac{x}{y} \times Y \quad Y = Y \quad Z = \frac{(1-x-y)}{y} \times Y$$

Unpacking Data

Receive Binary xyY Message

Convert to XYZ

Convert to CIE L*a*b*

Categorize input to Monk Category

Query Database for Matching product within threshold

```
2025-05-02 10:33:22,856 [INFO] Sent: 0
Received average xyY: [0.4275, 0.3854, 2.7239]
Recieved xyY values: [0.43, 0.39, 2.72]
XYZ Values Converted: [3.02, 2.72, 1.32]
Converted LAB Values: [18.90, 7.95, 14.20]
Monk Category: 9
```

$$L^\star = 116\, f(Y/Y_n) - 16,$$
$$a^\star = 500\left(f(X/X_n) - f(Y/Y_n)\right)$$
$$b^\star = 200\left(f(Y/Y_n) - f(Z/Z_n)\right)$$

where t is X/X_n, Y/Y_n, or Z/Z_n:

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3 \\ \frac{1}{3} t \delta^{-2} + \frac{4}{29} & \text{otherwise} \end{cases}$$
$$\delta = \frac{6}{29}$$

For Standard Illuminant D65:

$$X_n = 95.0489,$$
$$Y_n = 100,$$
$$Z_n = 108.8840$$

Expected L*a*b*

- CIE L*a*b* [18.838, 7.989, 15.207]

Calculated L*a*b*

- CIE L*a*b* [18.90, 7.95, 14.20]

Unpacking Data

Receive Binary xyY Message → Convert to XYZ → Convert to CIE L*a*b* → Categorize input to Monk Category → Query Database for Matching product within threshold

Reasoning

- Data is unevenly distributed throughout 10 Categories

Approach

- Queried CIELAB data based on classified monk categories
  - Reduces time complexity for targeted L*a*b* where # of products were significantly lower
- Calculated ΔE between target L*a*b* and database L*a*b* values

- Calculate ΔE between target and 796 Shades instead of 9846 Shades
- Only look at 8.084% of Shades

7	1333
8	$\boxed{796}$
9	103
10	39

Unpacking Data

Receive Binary xyY Message

Convert to XYZ

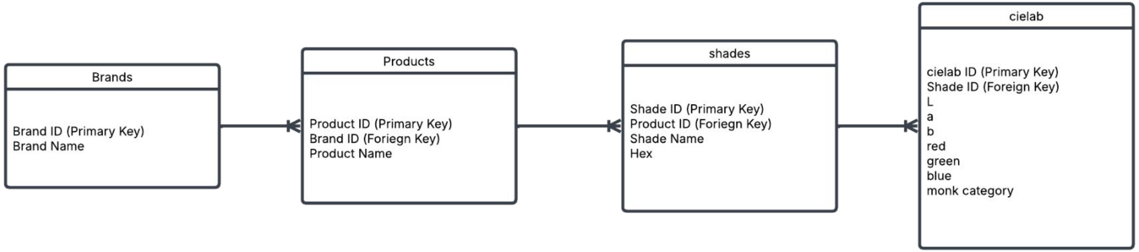
Convert to CIE L*a*b*

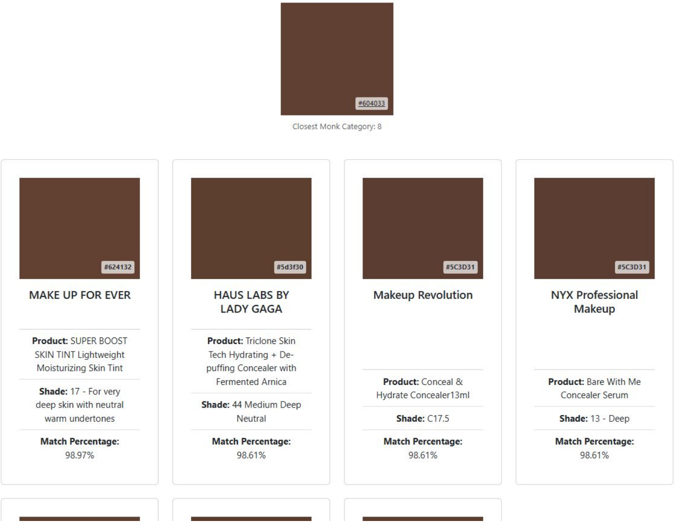
Categorize input to Monk Category

Query Database for Matching product within threshold

Approach

- Query Database for Brands & Product information where `cielab.shade_id = shade_id` foreign key from CIELAB table
- Display ΔE as Percentage for conceptual understanding





Color Accuracy Results

Monk	Accuracy Percentage without Lighting	Accuracy Percentage with Lighting
1	73.82%	63.59%
2	72.39%	75.35%
3	74.85%	75.32%
4	71.37%	71.70%
5	70.36%	70.55%
6	77.51%	76.69%
7	84.17%	84.08%
8	89.59%	90.62%
9	95.10%	96.67
10	94.87%	95.75%

5 Trials with & without lighting per each color

- Darkest Samples (Monk 7 - 10) had higher accuracy rates of 89 - 96%
- Lighter Samples (Monk 1 - 6) had lower accuracy 60 - 75%
- Slight improvements in accuracy with lighting adjustments

OPT4048 XYZ Sensor Limitations

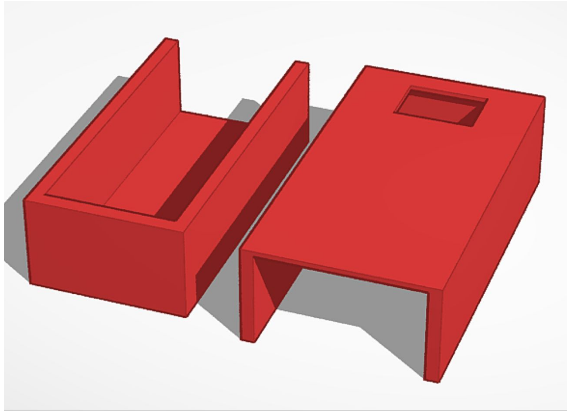
- Higher accuracy in darker conditions due to semi logarithmic output & automatic range selection
- Switches to lower full scale range for low light, provides finer resolution

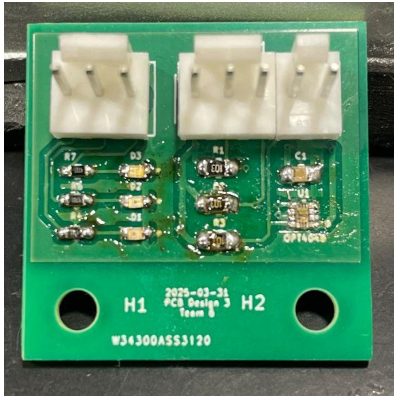
Conclusion

Successes & Challenges

Successes

- Separation of subsystems on PCBs allowed for individual testing
- Fixing our LDO so that it would not continuously short our circuit
- Building an enclosure for our device
- BLE communication between ESP32 & React client & Python server





Successes & Challenges

Challenges

- No output received from ESP32 via Serial Monitor
- Inconsistent power delivery to ESP32 GPIO pins
- Color sensor struggles to detect real-world object tones accurately
- Color sensor performs better with digital screen colors than physical surfaces

Daniel Balmaceda *9 days ago* TI__Expert 3810 points

Hi Shriya,

I am not able to access the code template from the link you have sent. Could you send a test log for the tests that were done in the different lighting conditions and distances?

As for your application, the OPT4048 can only detect the color of light that is emitted onto the sensing area. From your description, it seems as though you are trying to use the sensor to detect the color of an object (In this case human skin tone), which the sensor is not capable of doing on its own.

Conclusion

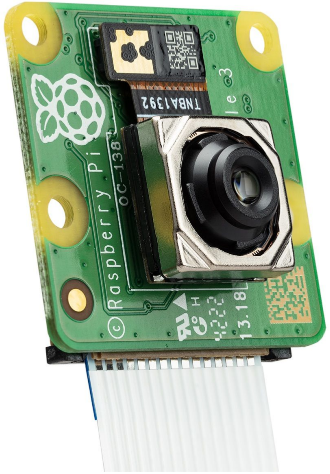
What We Learned

- Python BLE communication
- Subsystem separation improved debugging and testing
- Footprint mismatches can cause major hardware issues
- Datasheets don’t always match real world performance
- Lighting quality is critical for accurate color sensing

Conclusion

What We Would Change

- Use a sensor optimized for skin/surface color detection
- Incorporate a camera for consistent tone capture
- Design our PCBs to be more compact and the device to the more user friendly



What Are The Next Steps & Recommendations

- Explore sensors/cameras optimized for reflective surface scanning
- Add real time feedback, previews, and calibration tools to the website
- Develop a compact, all in one handheld hardware solution
- Market this as a much better colorimeter device for beauty stores like Sephora and Ulta to use on their clients for a more accurate shade match

Citations

[1] P. Goon, C. Banfield, O. Bello, and N. J. Levell, “Skin cancers in skin types IV–VI: Does the Fitzpatrick scale give a false sense of security?,” *Skin Health and Disease*, vol. 1, no. 3, Jun. 2021, doi: https://doi.org/10.1002/ski2.40.

[2] HTML Color Codes, "HTML Color Chart," Available: https://htmlcolorcodes.com/color-chart/.

[3] Bold Hue, "Bold Hue – Find Your Perfect Color Match," Available: https://www.boldhue.com/.

[4] Texas Instruments. “Choosing Between RGB and XYZ Color Sensors for Adaptive Lighting Adjustments.” Accessed: Mar. 06, 2025. [Online]. Available: https://www.ti.com/lit/ab/sboa567/sboa567.pdf?ts=1740975775684&ref_url=https%253A%252F%252Fwww.google.com.mx%252F

[5] C. Heldreth, E. P. Monk, A. T. Clark, S. Ricco, C. Schumann, and Xango Eyee, “Which Skin Tone Measures are the Most Inclusive? An Investigation of Skin Tone Measures for Artificial Intelligence.,” *ACM Journal on Responsible Computing*, Nov. 2023, doi: https://doi.org/10.1145/3632120.

[6] Wikipedia Contributors, “Fitzpatrick scale,” *Wikipedia*, Oct. 22, 2019. https://en.wikipedia.org/wiki/Fitzpatrick_scale

[7] P. Ken. HZDG, “What Is CIELAB color space?,” *Hunterlab.com*, 2022. https://www.hunterlab.com/blog/what-is-cielab-color-space/

[8] “Convert XYZ Color to L*a*b*,” *Mathworks.com*, 2025. https://www.mathworks.com/help/images/ref/xyz2lab.html (accessed Mar. 06, 2025).

[9] T. Fujiwara. “Color space conversion (3),” *Sakura.ne.jp*, 2019. https://fujiwaratko.sakura.ne.jp/infosci/colorspace/colorspace3_e.html (accessed Mar. 06, 2025).

[10] openoximetry.org, “Skin Color Quantification - OpenOximetry,” *OpenOximetry*, Sep. 13, 2024. https://openoximetry.org/skin-color-quantification/

[11] “Sensor Instruments,” *www.sensorinstruments.de*. https://www.sensorinstruments.de/whatiswhat.php?subpage=11&language=en

[12] M. Ellis. “Skin Tone Research @ Google,” *skintone.google*. https://skintone.google/get-started

[13] “Brief Explanation of delta E or delta E*,” *Hunterlab*, Jul. 20, 2022. https://support.hunterlab.com/hc/en-us/articles/203023559-Brief-Explanation-of-delta-E-or-delta-E

[14] Association for Computing Machinery, "*ACM code of ethics and professional conduct*". [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: Feb. 13, 2025].

[15] Institute of Electrical and Electronics Engineers. *“IEEE Code of Ethics”*.[Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: Feb. 13. 2025]

[16] Wikipedia contributors, “Color space,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Color_space . [Accessed: May 2, 2025].

[17] Wikipedia contributors, “Socket.IO,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Socket.IO . [Accessed: May 2, 2025].

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN