Circadian Lighting System

Areen Patil
iMTech CSE
IMT2023013
areen.patil@iiitb.ac.in

Nipun Verma
iMTech ECE
IMT2023591
nipun.verma@iiitb.ac.in

Unnath Chittimalla
iMTech ECE
IMT2023620
unnath.chittimalla@iiitb.ac.in

Prakrititz Borah
iMTech CSE
IMT2023547
prakrititz.borah@iiitb.ac.in

***Abstract*—In recent years, there has been a surge of interest in human-centric lighting (HCL). The key reason behind this growing interest is the advancements in technology, particularly LED lighting. LED technology opens up numerous possibilities that were previously unavailable, allowing us to tune both the spectrum and intensity of lighting. This flexibility makes LEDs an excellent foundation for human-centric lighting. Human-centric lighting should be more than just dimmable—it must be tunable. By combining hardware and software, lighting manufacturers now offer solutions that let users adjust and program both light intensity and spectrum. This adaptability is crucial for creating environments that support our natural circadian rhythms. We will walk through one of the possible implementations of circadian lighting here.**

***Index Terms*—circadian lighting, human centric lighting**

I. The History of Circadian Rhythms and Human Exposure

Before the industrial revolution, humans were exposed to minimal light at night. Technological advances have introduced new sources of light into our world including electronic screens (40 lux), lighting in homes (100–300 lux), and residential street lamps (15 lux) (Gaston et al., 2013). In contrast, the light from a full moon only emits 0.1–0.3 lux (Gaston et al., 2013). Indeed, 99% of Americans experience significant light pollution, defined as the alteration of natural nighttime lighting conditions (Falchi et al., 2016) With the advent of artificial lighting, especially in the last century, our exposure to natural light has drastically decreased. People started spending more time indoors, relying on artificial light that does not mimic the dynamic nature of sunlight. This shift has had profound effects on our circadian rhythms, leading to disruptions that impact mental and physical health. For example, having lights on in living spaces after dark delays the release of melatonin and may contribute to trouble sleeping (Gooley et al., 2011). Additionally, the wide prevalence of mobile phones with access to the Internet (i.e., smartphones), has led to light exposure throughout the day and night. Roughly 85% of US individuals own smartphones (Center, 2021), and many use their phones before bedtime (Hysing et al., 2015). The use of electronic screens delays the onset of sleep and decreases wakefulness the following day (Smotek et al., 2020).

II. What are Circadian Rhythms?

Circadian rhythms are 24-hour cycles that regulate various bodily functions, including sleep, wakefulness, hormone release, and metabolism. These rhythms are primarily influenced by the natural light-dark cycle of the environment. The brain's "master clock", located in the suprachiasmatic nucleus (SCN) of the hypothalamus, helps synchronize these rhythms with the day and night cycle. During the day, light signals the body to be awake and active, while darkness at night triggers the release of melatonin, a hormone that promotes sleep. This natural cycle ensures that bodily functions like sleep, digestion, and hormone production occur at the right times. However, the widespread use of artificial lighting, especially since the 19th century, has disrupted this natural synchronization. Artificial light, particularly at night, can confuse the circadian system, leading to a misalignment between our internal clocks and the external environment. This disruption can cause sleep disorders, metabolic issues, and increased susceptibility to mental health problems like depression. In environments where people are exposed to artificial light at odd hours, such as in shift work or in regions with long periods of darkness, the natural circadian rhythms can be severely disrupted, leading to various health problems.

III. One Month Progress

A. Project Overview:

This project aims to create a **circadian lighting system** that adapts to the human body's natural rhythm by adjusting lighting in a room based on the time of day. The system will rely on a **lookup table** that maps each hour to the ideal color temperature and light intensity, ensuring the lighting in the room syncs with human circadian rhythms, even accounting for current lighting conditions.

B. Key Components

1. Lookup Table (Time Zone Specific):

- **Time of Day:** Maps each hour to an ideal color temperature and intensity.
- **Color Temperature:** Reflects optimal light for circadian alignment at each hour.
- **Intensity:** Derived from the sun's position data for the specific time zone to maintain appropriate light levels.

Circadian Stimuli (CS): A metric used to quantify how light affects the circadian system. It regulates sleep-wake cycles, mood, and alertness.

Time of Day	Lighting Intervention	lambdamax (nm)	EV (lux)	CS
6:00 a.m.-8:00 a.m.	Cool Blue Light	480	100	0.50
8:00 a.m.-10:00 a.m.	Cool Blue Light	480	200	0.60
10:00 a.m.-12:00 p.m.	Bright White Light	6500K	N/A	0.45
12:00 p.m.-1:30 p.m.	Natural White Light	5500K	N/A	0.35
1:30 p.m.-3:00 p.m.	Warm White Light	3000K	N/A	0.20
3:00 p.m.-5:00 p.m.	Warm Red Light	630	50	0.10
5:00 p.m.-7:00 p.m.	Dim Red Light	630	30	0.05

TABLE I
LOOKUP TABLE FOR CIRCADIAN LIGHTING

2. Sensors:

Note: Sensors are yet to be implemented.

- **Light Intensity Sensor:** Detects current room light intensity.
- **Ambient Light Sensor:** Measures the color temperature of existing light.
- **Time Zone Data Input:** Provides sun position data to calculate required temperature.

IV. ALGORITHM OVERVIEW

A. Lighting Adjustment:

Once the sensors are configured, the system compares the detected light intensity with the lookup table's ideal intensity. If the room's current light intensity is lower than the ideal, a delta amount of light is added to reach the target intensity. The algorithm converts the desired color temperature and intensity into RGB values, suitable for driving LEDs.

B. Converting Color Temperature to RGB:

Using **Planck's Law** and the **CIE 1931 XYZ color matching functions**, we can map color temperature to RGB values for LEDs.

- **Planck's Law:** Describes the intensity of light emitted by a black body at a given wavelength and temperature.
- **CIE 1931 XYZ Model:** Represents how humans perceive color based on different wavelengths.
- **RGB Conversion:** XYZ values are converted to RGB using a transformation matrix, yielding RGB values between 0-255 for LED control.

PROPOSED METHOD

Step 1 - Calculate Spectral Radiance (Planck's Law):

$$L(\lambda, T) = \frac{2hc^2}{\lambda^5} \times \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1}$$

Where:

- λ is the wavelength
- T is the color temperature in Kelvin
- h is Planck's constant
- c is the speed of light
- k_B is Boltzmann's constant

Step 2 - Integrate with CIE 1931 Color Matching Functions:

Integrate the spectral radiance over the visible spectrum (380nm to 780nm) with the CIE 1931 functions to derive the XYZ values:

$$X = \int L(\lambda, T) \cdot x(\lambda) d\lambda, \quad Y = \int L(\lambda, T) \cdot y(\lambda) d\lambda$$

$$Z = \int L(\lambda, T) \cdot z(\lambda) d\lambda$$

Step 3 - Convert XYZ to RGB Using a Matrix Transformation:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2406 & -0.9689 & 0.0557 \\ -1.5372 & 1.8758 & -0.2040 \\ -0.4986 & 0.0415 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Normalize the resulting RGB values to the 0-255 range for display or LED lighting control.

C. PRACTICAL APPLICATION FOR LEDS

Once the RGB values are determined, they can be directly applied to control the intensity of red, green, and blue LEDs to reproduce the desired color corresponding to the input color temperature.

D. EXAMPLE: CONVERTING 6500K TO RGB

- Calculate Spectral Radiance: Use Planck's Law to determine the intensity of light across the visible spectrum.
- Integrate with CIE 1931 Functions: Combine this spectral data with the color matching functions to derive the XYZ values.
- Transform XYZ to RGB: Apply the transformation matrix to obtain RGB values.

For 6500K, the process might yield RGB values close to [255, 255, 255], which represents white light—consistent with what we expect from daylight.

Since color temperature is an abstract value and we do not know of any hardware that can sense this quantity directly, we might have to use simple machine learning models like SVM, or simple classification models that at least tell us the range of color temperatures based on the data's location, sun position, weather, etc. This color temperature is useful in the case where we want to adjust our lookup tables to be specific to that location.

The algorithm for displaying a specified color temperature to an LED is based on an approximation and may be computationally expensive to run on hardware like the Arduino UNO. So, the algorithm is run on a computer, and then the generated lookup tables are copied to the microcontroller. However, using more powerful microcontrollers like the Raspberry Pi could change this case.

The algorithm converts color temperature to RGB by first calculating the spectral radiance using Planck's Law, then integrating it with the CIE 1931 color matching functions to get XYZ values, which are finally transformed into RGB using a standard matrix. This approach ensures accurate color representation for LEDs based on human color perception.

V. Naive Implementation in Tinkercad (As an example)

A brief idea of how the basic structure or working can be found here in a Tinker CAD Simulation where 24 hours of the day are simulated in 24 seconds with hardcoded values. This can be found in the Links section.

VI. Hardware Setup for Raspberry Pi 3B+

In this section, we will outline the procedure to integrate the code with the Raspberry Pi 3B+ hardware, allowing us to control an RGB LED to simulate circadian lighting. The RGB LED will be connected to the Raspberry Pi's GPIO pins, and Pulse Width Modulation (PWM) will be used to adjust the intensity of each color channel (Red, Green, Blue) according to the desired color temperature.

A. Wiring the RGB LED to the Raspberry Pi

The RGB LED should be connected to the Raspberry Pi's GPIO pins as follows:

- The Red (R) pin of the LED is connected to GPIO pin 12 (PWM capable).
- The Green (G) pin of the LED is connected to GPIO pin 13 (PWM capable).
- The Blue (B) pin of the LED is connected to GPIO pin 19 (PWM capable).
- The common ground pin of the LED is connected to one of the Raspberry Pi's ground (GND) pins.

This configuration allows the use of hardware PWM on the Raspberry Pi to control the brightness of each color channel independently. By adjusting the duty cycle of the PWM signal, we can control the intensity of each color and simulate different color temperatures throughout the day.

B. Software Setup

The code uses Python's `RPi.GPIO` library to interact with the GPIO pins on the Raspberry Pi. Below is the high-level flow of the code that controls the LED based on the desired color temperature.

1) The GPIO pins for the RGB LED are set up using the `RPi.GPIO` library, and hardware PWM is initialized for each pin with a frequency of 100 Hz.
2) A CSV file containing the CIE 1931 color matching functions is loaded, and the wavelength data is extracted.
3) The `ColorTemperatureConverter` class converts a given color temperature (in Kelvin) to the corresponding RGB values using the following steps:
   a) The spectral radiance for the given color temperature is calculated using Planck's law.
   b) The XYZ color values are obtained by integrating the spectral radiance with the CIE 1931 color matching functions.
   c) The XYZ values are converted to the sRGB color space using a transformation matrix.
   d) The resulting RGB values are normalized and scaled to the [0, 255] range.

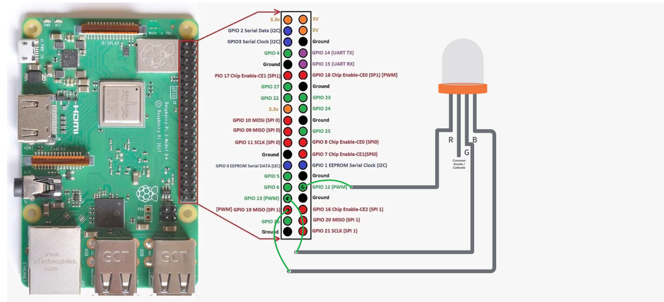

Fig. 1. Schematics

4) The calculated RGB values are applied to the RGB LED by updating the duty cycle of the PWM signals for each color channel.
5) The LED is updated every 5 minutes, simulating the changes in color temperature over time.

VII. Design

Above is the Basic Schematic that controls the RGB LED connected to the Raspberry Pi 3B+ (220ohm resistors to be used between GPIO pins and RGB LED pins):

We plan to connect this code with dynamic lookup tables based on location and also set customized circadian routines via a web front-end in the future. The code can be found in the GitHub Repository mentioned in the Links section.

Links

- TinkerCAD simulation: simulation
- GitHub link: github
- CIE 1931 Specification Dataset: dataset
- Inspiration for Color Temperature to RGB Logic: inspiration