ECE 445
Senior Design Laboratory
Final Report
FALL 2024

Ambient Lighting System

Team 41
Anusha Adira (adira2@illinois.edu)
Manushri Dilipkumar (md38@illinois.edu)
Chinmayee Kelkar (ckelkar2@illinois.edu)

TA: Chentai (Seven) Yuan
Professor: Kejie Fang

December 11, 2024

Abstract

This project introduces an intelligent lighting system that automates and personalizes the lighting experience by responding to sound and heart rate. Using LED strips that can display ten colors and adjust brightness, the system reacts to background noise and heart rate data. The audio subsystem detects environmental sounds, triggering real-time color transitions, while the heart rate monitor adjusts brightness based on the user's heart rate. This adaptive system enhances the ambiance for activities like watching a movie or listening to music, with responses occurring within 50ms for color changes and 100ms for brightness adjustments, reducing the need for manual intervention.

Contents

1 Introduction

1.1 Problem

In many environments, the ambiance is heavily influenced by lighting, which often requires manual adjustment to match the mood of the space. This can be inconvenient and limits personalization. What if we had a system that could track how you are feeling and adjust the lighting system accordingly? We propose an individual lighting experience that acts as a dynamic lighting system, reacting to sound and heart rate to provide a personalized, immersive environment. This system would eliminate the need for manual intervention, offering a more cohesive ambiance that changes based on both noise and the user's emotional state.

For example, if a user is watching an action movie or playing music, the system will synchronize lighting with the intensity of the scene or sound. In addition, the system adjusts the brightness based on the user's heart rate, creating a unique experience tailored to their mood and activity.

1.2 Solution

Our project proposes the development of an intelligent lighting system that connects LED strips, which can be placed behind a TV, painting, or near a speaker. The system automatically synchronizes with the background noise of the user's activity, while also adjusting intensity based on the user's heart rate. This enhances the user experience by providing adaptive lighting that is highly personal and responsive.

At a high level, we have an audio system that collects background audio and sends signals to change the color of the LED strip. Additionally, a heart monitor system connects to the circuit via Wi-Fi connection and sends signals to adjust the intensity of the LED strip—brighter for higher heart rates and dimmer for lower heart rates.

1.3 Visual Aid

Figure 1 shows at a high level what our project does. It takes the microphone output (environment) and the heart rate monitor output (individual) and sends it to the computation system to then have a color outputted to the LED strips.

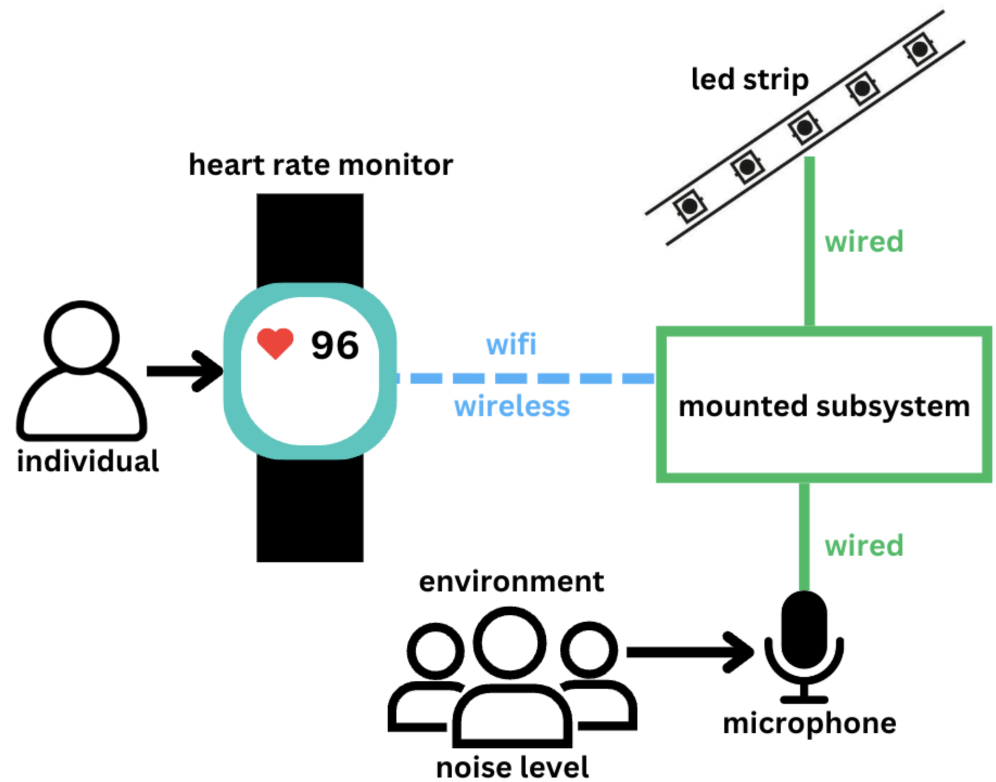

Figure 1: Visual Aid

1.4 High-Level Requirements

- The system must detect and respond to changes in noise levels, adjusting the LED color within 50ms.
- The heart rate monitor must accurately transmit data, allowing for brightness adjustments within 100ms of detecting a change in heart rate.
- The LED strip must support a full range of 10 colors and brightness levels from 200 lumens to 450 lumens, allowing for highly customizable lighting experiences based on sound and heart rate.

2 Design

2.1 Block Diagram

Figure 2 consists of the high level block diagram. This section shows all of the subsystems and their interactions with other subsystems.

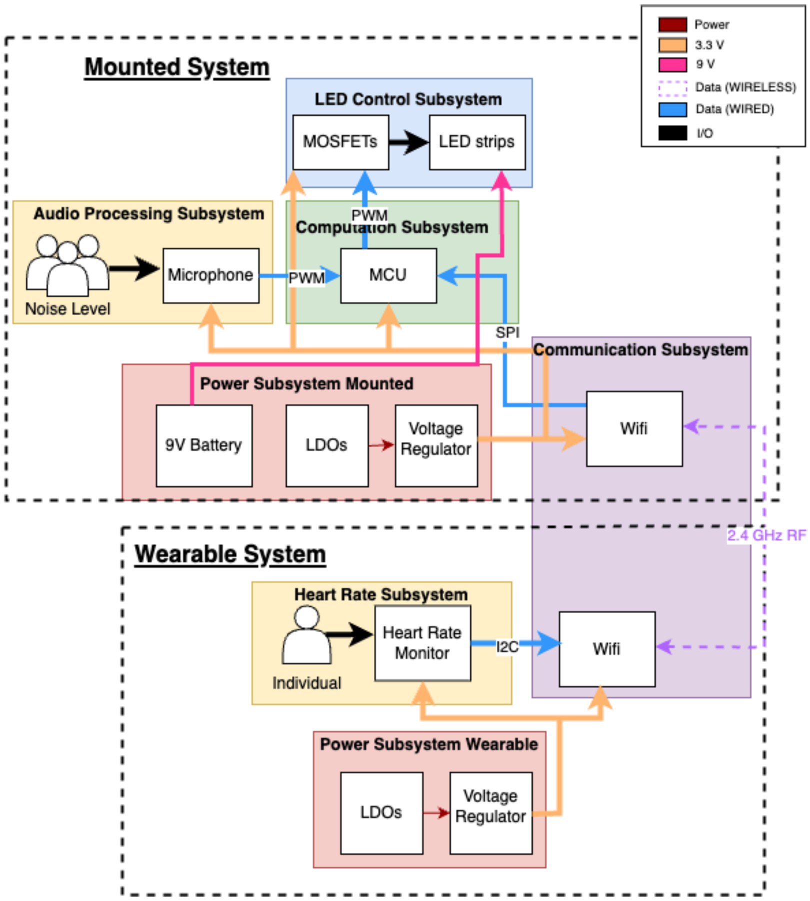

Figure 2: Block Diagram

For overall circuit schematic diagrams, look at Appendix B.

2.2 Audio Processing Subsystem

2.2.1 Design procedure

The Audio Processing Subsystem is responsible for capturing and conditioning the audio signal to ensure it is suitable for further processing in the computation subsystem. This involves selecting and integrating components for signal acquisition, amplification, and optimization. For this subsystem, we use an Electret Microphone Amplifier - MAX9814 with Auto Gain Control (AGC), chosen for its robust performance and adaptability in handling varying input sound levels [1].

Some alternative approaches would be to use a regular Electret Microphone or a MEMS microphone and a separate Amplifier, but we thought this complication was unnecessary, further proving that we should stick with the MAX9814 with AGC module [2]. Along with the simplicity, this design is quite reliable, ensuring that we are getting clear audio with little distortion so that we can process these audio signals easily. In addition, the module is quite compact, helping decrease our power consumption and making it easy to place on the PCB [3]. There were no design issues regarding the microphone module.

2.2.2 Design details

Our circuit schematic is quite simple since we are just allowing for connection in the MAX9814 module. Figure 3 shows the microphone module schematic.

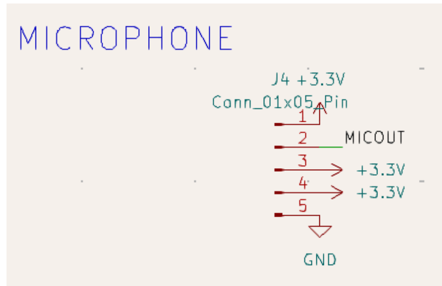

Figure 3: Microphone module

According to the Datasheet, the “OUT” pin is the only pin connected to the microphone output; the “GND” pin is connected to GND and the rest are connected to power [3].
To adjust the gain on the MAX9184, we can either connect the Gain pin to ground, Vdd, or floating to increase it by 20, 40, or 60 dB. When connected to Vdd, there is a gain of 40dB. A gain of 40 dB ensures that even faint audio signals (e.g., soft speech or distant sounds) are amplified sufficiently to fall within the operational range of our system. This is essential for our audio processing system to track the audio and send it to the computation subsystem.

2.3 Heart Rate Monitoring Subsystem

2.3.1 Design procedure

The heart rate monitoring subsystem is designed to measure the users heart rate in real time and send this to the MCU for processing. The system utilizes a heart rate sensor and the sensor is transmitting through WiFi to the MCU. To achieve the required accuracy within ±10 beats per minute, calibration is performed by comparing the sensor’s output to a reliable device such as an Apple Watch. We needed the heart rate sensor, so we did not come up with any alternatives. In addition we did not face any design challenges.

2.3.2 Design details

Figure 4 shows the Heart Rate Sensor Schematic.

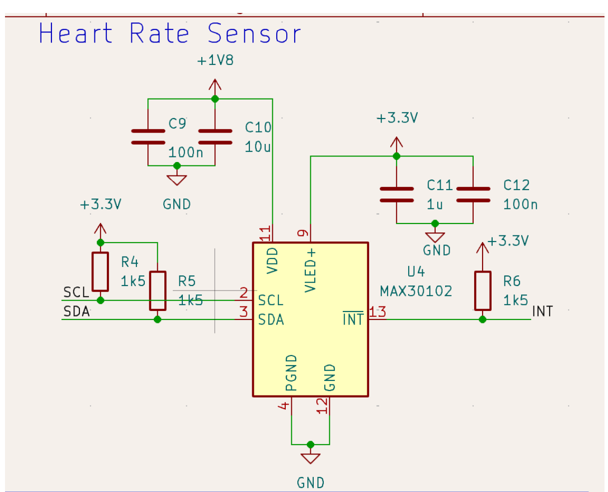

Figure 4: Heart Rate Sensor Schematic

This subsystem transmits heart rate data to the MCU, which combines it with audio data to adjust the brightness of the LED lights. It will also receive power to the heart rate monitor from the power subsystem.

The heart rate sensor transmitted the raw infrared values via I2C communication [4]. Using built in libraries, we were able to abstract the heart rate and calculate the moving average so that the average heart rate is accurate to the user's current heart rate. The libraries abstract the heart by doing periodic waveform analysis and getting the peak to peak, and the heart rate is then equal to $\frac{60}{T}$, in which T is the time between the peak to peaks.

2.4 LED Control Subsystem

2.4.1 Design procedure

Our LED Control system takes the color and brightness output from the computation subsystem and then outputs the RGB values to the LED strip [5]. It manages the color and brightness of the LED strips. The LED driver receives control signals from the MCU, which processes both the audio input and heart rate data. Manual brightness settings are used to control the brightness, based on heart rate, while color changes are driven by the audio analysis. This allows for fine-tuned, real-time control over the LED display, making it responsive to both environmental sounds and physiological signals. During the design phase, alternatives such as a purely audio-driven system or an exclusively heart-rate-based control mechanism were considered but were found to be less versatile. The chosen design strikes a balance, leveraging both audio and physiological inputs to achieve dynamic and engaging LED behavior.

2.4.2 Design details

Figure 5 shows the LED subsystem schematic with three MOSFETs and a connector pin for the LED strips.

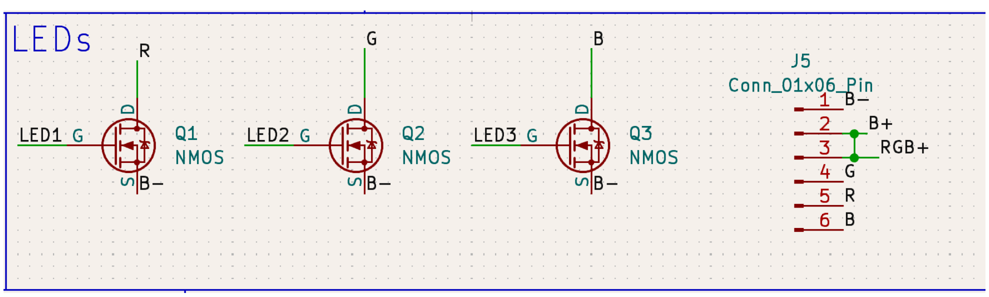

Figure 5: LED Subsystem

The RGB values are inputted to the LEDs as shown in the diagram above. The led strip requires a battery, as the 3.3V output isn't sufficient to power the entire strip. Since there is a higher current draw, the MOSFETs are required to provide it.

2.5 Power Management Subsystem

2.5.1 Design procedure

We need a 3.3V and 1.8V source for each of our different subsystems, so we decided to divide the voltage down from the input from the USB-C input. The alternative approach would be to power the subsystems using a battery or an AC/DC converter, but this seemed to be better as we could use the same circuitry for both the mounted and the wearable system initially [6]. Another design alternative considered was the use of dedicated voltage regulators for each subsystem to improve efficiency, but this was deemed unnecessary given the compact design requirements [7]. By using the USB-C input for power distribution, we ensured consistent and reliable operation across multiple configurations.

2.5.2 Design details

Figure 6 shows the schematic for the Voltage Regulator.

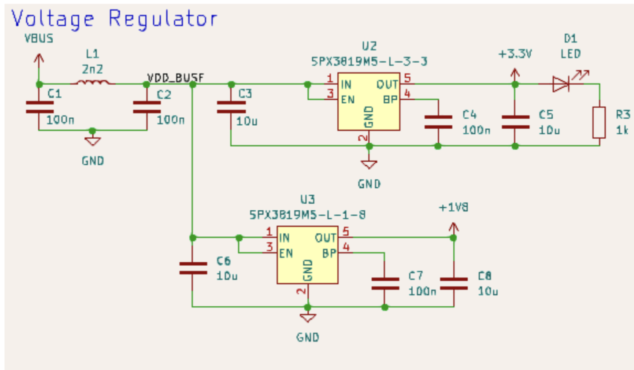

Figure 6: Voltage Regulator Schematic

Two LDO's were chosen because they divided down the voltage to 3.3V and 1.8V. The supporting components were based on the datasheets[7] typical application circuit. The output of these two LDO's were used to power the circuitry.

2.6 Communication Subsystem

2.6.1 Design procedure

The communication between the two ESP32's is via Wi-Fi. This was chosen over Bluetooth as we had better communication and there was less latency when we chose to use Wi-Fi [8]. The communication subsystem is vital to the system as we needed to find a way to communicate the heart rate data to the computation subsystem since the brightness is controlled by the heart rate. Some alternatives for this would be to use a different ESP32 module like the ESP32 Wroom. This led to one of our biggest design issues since the ESP32 we used required a lot of external powering instead of the Wroom which had everything built into it.

2.6.2 Design details

The communication subsystem takes the data from the heart rate monitor and communicates that with the MCU in the mounted subsystem. It gets power from the wearable power subsystem. This subsystem does not directly interact with the mounted power subsystem, the LED driver, or the audio processing subsystem. Figure 7 shows the antenna schematic.

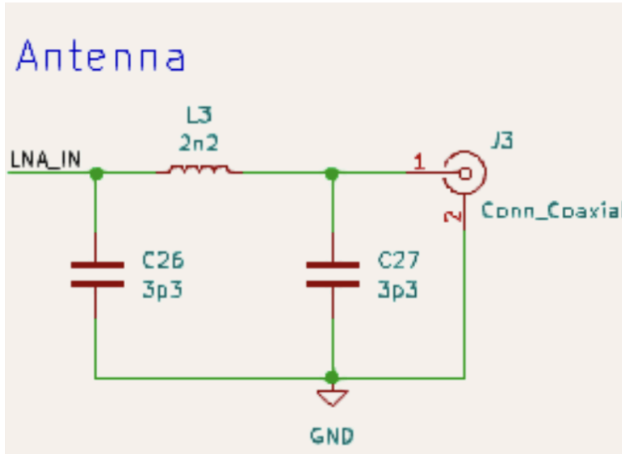

Figure 7: Antenna Schematic

The antenna is required for the communication subsystem as the ESP32-C3 does not have a built in antenna. The surrounding components were based on the typical circuit in the datasheet [8].

2.7 Computation Subsystem

2.7.1 Design procedure

This subsystem takes the audio information from the audio processing subsystem and the heart rate information from the wearable subsystem and does the necessary calculations to convert that into a color and brightness values that that LED system will read so the LED strip changes to the appropriate color at the appropriate time. The programming circuitry is required to program the ESP32 with the correct code, so that is a requirement for the computation subsystem. Like in the previous subsystem, some alternatives for this would be to use a different ESP32 module like the ESP32 Wroom. This led to one of our biggest design issues since the ESP32 we used required a lot of external powering instead of the Wroom which had everything built into it.

2.7.2 Design details

Figure 8 shows the MCU schematic.

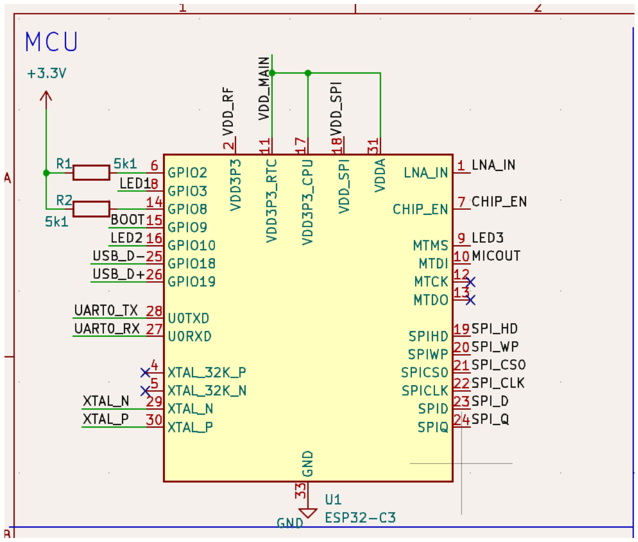

Figure 8: MCU Schematic

This subsystem consists of an MCU. It is powered by the mounted power subsystem, takes in input from the audio processing subsystem, the communication subsystem and outputs that data to the LED control system [8]. It doesn't interact with the wearable power subsystem or the heart rate monitor.

2.8 Tolerance Analysis

Data Size and Transmission Feasibility:

- Data Size for Heart Rate: Assuming the heart rate value is an integer, it requires 4 bytes (32 bits) for transmission.
- Wi-Fi Bandwidth: At 1 Mbps, the Wi-Fi module transmits 1 Megabit per second or 1,000,000 bits per second.

Transmission Time Calculation:

- Time to Transmit 32 bits: The time to transmit the 32-bit integer is: Transmission Time=1,000,000 bits/sec32 bits=32×10−6 sec=32 microseconds.
- This time is negligible compared to the requirement of 50 ms for transmission, so the transmission rate is well within the allowable tolerance.

Transmission Latency and Buffering:

- Wi-Fi Latency: Wi-Fi has a typical latency of 10 ms to 20 ms depending on environmental factors like interference. To account for worst-case scenarios, we assume the maximum latency of 20 ms.
- Total Latency: The total communication delay will be the sum of the Wi-Fi transmission time and any additional system processing or buffering. Since the transmission time is negligible (32 microseconds), the total latency is effectively dominated by the Wi-Fi latency, which is 20 ms.

- This leaves a 30 ms buffer for processing at the MCU, well within the requirement of 50 ms.

Power Supply Tolerance:

- The Wi-Fi transmitter operates at 3.3V, which is integrated into the ESP32, with a tolerance of ±0.1V. Power fluctuations could cause performance degradation or interruptions.
- Power Tolerance: The voltage regulation circuitry needs to ensure that the Wi-Fi module, integrated into the ESP32, is supplied with a stable 3.3V ± 0.1V. This is achievable with standard voltage regulators that maintain accuracy within 1-2% of the nominal voltage. Hence, the power supply tolerance will not pose a significant challenge.

Mathematical Analysis of Time Constraint:

Total time, t_{total}, for data sampling, transmission, and processing uses this equation:

$$t_{total} = t_s + t_{bt} + t_{proc} \tag{1}$$

Define the variables:

- t_s= sampling interval = 100 ms,
- t_{bt} = Wi-Fi transmission time (including latency) = 20 ms,
- t_{proc}= MCU processing time = 10 ms (worst case).

Adding all of these variables up, we get that the total time is 130 ms. Since the system must complete the heart rate monitoring cycle within 150 ms, we know that the timing requirements are feasibly met, with a 20 ms buffer.

By analyzing the heart rate data transmission subsystem, we can see that the **Wi-Fi transmission time**, **latency**, and **processing time** all fit well within the given **150 ms constraint**. The power supply requirements also fall within standard tolerances. Hence, with the given design parameters, the subsystem can **feasibly meet** its performance requirements, and the risk of failure due to timing or power issues is minimal.

3 Design Verification

3.1 Audio Processing Subsystem

3.1.1 Requirements

This subsystem was responsible for collecting the background noise around the user. The main part is the microphone which will read this background noise and send it to the computation subsystem. The first requirement is that this subsystem must capture the audio within a 20-decibel range of the actual sound. The second requirement is that the lights must change according to the sound ensuring an accurate environmental or video audio representation.

3.1.2 Verifications

We can verify that this system is in a 20-decibel range by generating a sound of a known decibel level with a decibel meter and then outputting the result of this sensor to the computer. If it is within this range, we can ensure that this high-level requirement is met. Additionally, we can play a sound of 60dB, 80dB, and 100dB near the microphone and ensure that the lights are changing according to these different sound levels. Also see Appendix A Table 4 for the full table of Requirements & Verifications.

3.2 Heart Rate Monitoring Subsystem

3.2.1 Requirements

The heart rate monitoring system has 2 specific requirements to ensure its performance. The sensor must provide accurate measurements within a margin of 10 beats per minute of the users heart rate when it is compared to a reliable device such as a Fitbit or Apple Watch. Additionally, the system must transmit heart rate data to the MCU via a Wi-Fi connection within 150ms of detecting a change in heart rate. This is critical for ensuring its accuracy.

3.2.2 Verifications

Verifying this requires two steps. First, the accuracy of the heart rate sensor must be tested by comparing the printed readings with an Apple Watch and ensuring that they are within the 10 bpm range. Second, the data transmission is verified by measuring the time interval between when the heart rate is detected by the sensor and when it is sent to the mounted system. This can also be done manually at a rate of 100ms. Multiple trials are conducted to ensure that this works for a range of heartbeats and occurs within the 150ms threshold. Also see Appendix A Table 5 for the full table of Requirements & Verifications. Figure 9 below shows that the readings were in fact within the 10 bpm range.

Figure 9: Example of Heart Rate verification

3.3 LED Control Subsystem

3.3.1 Requirements

The LED control subsystem has two requirements to ensure it is functioning appropriately. This system must be capable of displaying 10 different colors with smooth transitions between the hues to create a visually engaging experience. Additionally, the LEDs must adjust their brightness levels in response to heart rates and reach a maximum brightness when the user's heart rate reaches 120 bpm.

3.3.2 Verifications

To verify this, the LED strip is manually observed and we must visually see the 10 colors. We can visually detect the distinct colors and ensure the transitions are smooth to the human eye. Second, the brightness is tested using a reliable heart rate monitoring device such as an Apple Watch. Once the user's heart rate hits 120 bpm which is cross-referenced by the Apple Watch, the system must reach its maximum LED brightness. Also see Appendix A Table 6 for the full table of Requirements & Verifications.

3.4 Power Management Subsystem

3.4.1 Requirements

The requirement for this system is that it must not draw more than 500mA of current to preserve the battery life. We also have to ensure that the battery is enough to power all the LEDs for at least an hour.

3.4.2 Verifications

We can verify this using a multimeter when the battery is connected to the LEDs and ensure it is not drawing too much current. We can verify that the consistent 3.3V is provided with a voltmeter and perform a test to ensure that the battery life lasts for at least 1 hour.

We wanted to see how long the batteries would power the LED to determine the battery life using the following equation [9]:

*Runtime (hours) = Battery Capacity (mAh) / (Current Draw (mA)) * Efficiency Factor)* (2)

We have a 9V battery with a 500 mAh capacity, and our led strip has a 5W power rating which can be converted to a 560 mA draw. The MOSFET has a listed efficiency of 90%. We can then calculate the runtime with these known variables:

Runtime (hours) = 500 *(mAh) /* (560 *(mA)) * 0.9) = 0.99 hours* (3)

We found that the runtime after calculations would be about 1 hour. This is quite a short runtime, but it is sufficient enough for the scope of this project. Also see Appendix A Table 7 for the full table of Requirements & Verifications.

3.5 Computation Subsystem

3.6.1 Requirements

The computation system is responsible for processing input data and ensuring accurate outputs for the lighting system. The subsystem must trigger a color change in the LEDs for every 5dB change in the background noise to ensure it is in sync with the media that the user is consuming. Additionally, for every 10 bpm change in heart rate, the brightness of the LEDs must adjust by 30 lumens which will keep the system in sync with the user.

3.6.2 Verifications

There are two key verifications to ensure this system is working properly. The background audio with varying dB levels is played into the microphone to confirm that the subsystem is changing accurately and triggers the correct color changes in the lights. Hardcoded bpm values are sent to the computation subsystem to simulate a change in heart rate and the resulting brightness levels are monitored to ensure they adjust by 30 lumens for every 10 bpm change. This will ensure accurate lighting results. Figure 10 shows the relationship between Heart Rate and Brightness.

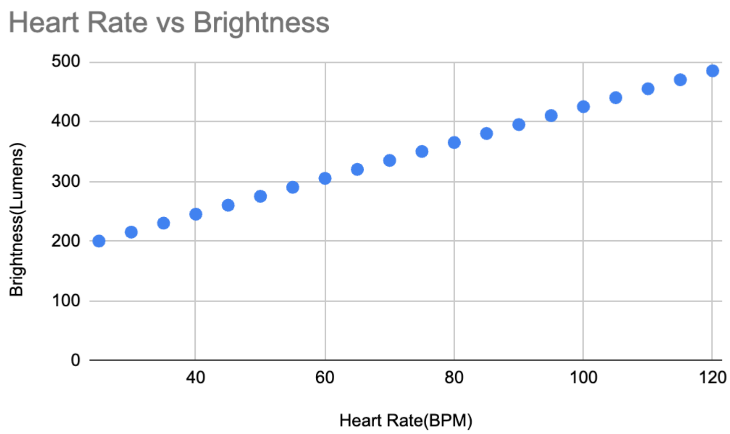

Figure 10: Heart Rate vs. Brightness graph

We calculated the lumens brightness level in the graph by this equation:

$$Maximum\ Brightness = LED\ Count \cdot Lumens \tag{4}$$

After counting the number of LEDs on the strip(41) and multiplying the lumens per LED(11 Lumens), we see that the maximum brightness is 450 lumens. Since the amount of power applied to the LEDs is linearly proportional to the brightness in lumens, we can see that at half power the brightness is at 200 lumens. Also see Appendix A Table 9 for the full table of Requirements & Verifications.

4 Costs & Schedule

4.1 Parts

Table 1: Parts Cost Analysis

Description	Part #	Cost per unit	Amount	Link
Pulse Oximeter Sensor	MAX30105	$12.70	1	Amazon
MCU with integrated Wi-Fi and Bluetooth	ESP32-C3	$1.00	2	Digikey
LED Strip	WS2812B	$27.99	1	Amazon
Electret Microphone	MAX9814	$7.95	1	Adafruit
Antenna	ANT-W63RP C1-UFL-200	$5.72	2	Digikey
2M Flash	W25Q16JVU	$0.61	2	Digikey

	XIQ TR			
Crystal	800G331	$0.75	4	Adafruit
ESD Protection Diode	DESD12VL1 BAQ-7	$0.22	4	Digikey
USB-C Connector	USB4125-GF -A-0190	$0.59	2	Digikey
Voltage Regulator	TPS62202DB VR	$1.21	6	Digikey
Switch - Switch Tactile	1825910-6	$0.10	7	ECE shop
Resistors - 330 Ω	0805	$0.07	7	ECE shop
Resistors - 1.5k Ω	0805	$0.07	10	ECE shop
Resistors - 5.1k Ω	0805	$0.07	15	ECE shop
Resistors - 22 Ω	0603	$0.07	4	ECE shop
Capacitors - 10pF	0603	$0.24	6	ECE shop
Capacitors - .1µF	0603	$0.24	24	ECE shop
Capacitors - 10 µF	0805	$0.24	18	ECE shop
Capacitors 1µF	75-562R5HK D10	$0.24	5	ECE shop
MOSFETs	SFF60P05M	$0	3	Self Service
Total Cost		$91.56		

4.2 Labor

Table 2: Labor Analysis

Name	Weekly Hours	Hourly Pay	Weeks	Cost (USD)
Anusha	15	21	12	3,780
Chinmayee	15	21	12	3,780
Manushri	15	21	12	3,780
Total	45	21	12	**11,340**

4.3 Total Cost

Sum of costs: \$11340 + \$91.56 = \$11431.56 total to produce this project

4.4 Schedule

Table 3: Schedule

Week	Anusha	Chinmayee	Manushri
Week of 10/7	-Finalize and review PCB design schematic	-Research microcontrollers and heart rate sensors	-Research sound detection
Week of 10/14	-Finalize subsystem schematics after getting feedback	-Order all necessary components including PCB, LED strip, heart rate sensors, Wi-Fi transmitters, and MCU	-Start with the basic setup of MCU programming
Week of 10/21	-Receive PCB and inspect to make sure there are no defects	-Begin the soldering process and assembling the Audio Processing and Heart Rate Monitor subsystems	-Test individual components for functionality
Week of 10/28	-Complete assembly of Audio Processing and Heart Rate Monitor subsystems	-Start assembling LED subsystem -Test initial MCU functionally for the audio data processing	-Conduct tests on individual subsystems and see if any areas need improvement -Begin working on Wi-Fi communication for heart rate data
Week of	-Integrate LED control	-Conduct unit test on	-Begin assembling the

11/4	with MCU	LED color and brightness adjustments	power management system both mountable and wearable
Week of 11/11	-Integrate Audio Processing, Heart Rate Monitor, and LED Control subsystems.	-Establish reliable Wi-Fi communication between wearable and mounted systems.	-Test data transmission and debug and resolve integration issues
Week of 11/18	- Perform comprehensive system integration tests.	- Ensure real-time responsiveness of lighting adjustments.	- Conduct user testing to gather feedback on system performance. - Begin refining the PCB layout based on testing results (if necessary).
Week of 11/25	Fall Break	Fall Break	Fall Break
Week of 12/2	- Develop the final presentation, including slides and demonstration scripts. - Prepare the final demo setup, ensuring all components are functioning correctly.	- Develop the final presentation, including slides and demonstration scripts. - Test with varying heart rate and sounds and ensure there are no issues	- Develop the final presentation, including slides and demonstration scripts. - Perform last-minute testing and troubleshooting.
Week of 12/9	Final presentation and final papers	Final presentation and final papers	Final presentation and final papers

5 Conclusion

5.1 Accomplishments

Overall our project was mostly successful. Although we did not get our PCB to work, the breadboard and dev kit were functioning successfully. The lights did change throughout the 10 colors and were visibly changing based on the beat of the music or the sound changes. The brightness levels were also able to change based on changes in heart rate from 60-120 bpm. The product worked successfully visually and created a customizable user lighting experience using the user's heart rate and background noise.

5.2 Uncertainties

Some uncertainties we had were that the heart rate monitor might not provide consistent readings which is something we struggled with. There are certain conditions that would also affect this such as varying moisture levels on the skin and the user's hand placement not being ideal on the heart rate monitor. The music had to be played quite close to the microphone for it to be read accurately which is something we can also improve upon in the future. Another uncertainty would be that we do not know how this system would react to extreme temperatures, physical activity, or any effect of accidental damage. These uncertainties highlight the areas that could require further testing and validation to ensure that this product works as it is intended to.

5.3 Ethical considerations

Our project involves the use of heart rate data and background noise, which raises ethical concerns about data privacy and user safety. To address these concerns, we adhere to the ACM Code of Ethics (Section 1.6) by ensuring that users' personal data is protected. We will avoid storing their data and will implement measures to prevent unauthorized access. Additionally, as required by the IEEE Code of Ethics (Section 1), we will inform users of potential risks and obtain explicit consent before any data is collected or used. Users will be made aware of how their data is processed with clear consent procedures in place and we will not store their data anywhere.

Accessibility is another ethical priority. According to the ACM Code of Ethics (Section 1.4), fairness and non-discrimination are essential, meaning the system must be accessible to individuals with disabilities. We will issue a warning before system use to mitigate risks posed by lights flashing or changing intensity, which could potentially harm users' eyes. Rapid light flickering or intense brightness can cause eye strain or migraines, so we will ensure the system minimizes these effects, keeping brightness transitions gradual and flicker rates safe. There will be a warning to all users before using this product to ensure their safety.

On the safety side, the wearable heart rate monitor will comply with the IEC 60601-1 standard to ensure user safety. The device will be designed to avoid overheating or malfunctioning, as these risks could harm users. Protecting users from excessive heat is critical, and careful testing will be conducted to verify device reliability.

Finally, we will provide users with an easy way to turn off the system if they experience discomfort from the lights. By following these ethical guidelines and safety standards, our project prioritizes user safety and privacy throughout the design process.

5.4 Future work

In the future, we hope to get our PCB to work successfully. We had some delays with specific parts not matching such as the USB connector and the crystal. Although our breadboard was working, we would hope to get the PCB to work to ensure a sleeker design. Our heart rate monitor was a touch sensor because of the way the chip was on the breadboard. We hope to make this design more user-friendly by putting this sensor on a wristwatch similar to a Fitbit or an Apple Watch. This way the user would not have to rest their finger on this sensor and would have more freedom to move. We could also move to make the enclosure for the mounted system sleeker as the box currently 3D printed was a bigger enclosure than needed.

References

[1] Analog Devices, Inc., *Microphone Amplifier with AGC and Low-Noise Microphone Bias*, Datasheet. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX9814.pdf.

[2] Analog Devices, Inc., *ADMP401 MEMS Microphone with Bottom Port and Analog Output*, Obsolete Datasheet. [Online]. Available: https://www.analog.com/media/en/technical-documentation/obsolete-data-sheets/ADMP401.pdf.

[3] Analog Devices, Inc., *General Description*, Datasheet. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/max9814.pdf.

[4] Analog Devices, Inc., *MAX30100 Pulse Oximeter and Heart-Rate Sensor IC*, Datasheet. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30100.pdf.

[5] Adafruit Industries, *WS2812B Intelligent Control LED*, Datasheet. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf.

[6] MEAN WELL USA, *IRM-05 Series 5W AC/DC Converter*, Datasheet. [Online]. Available: https://www.meanwellusa.com/upload/pdf/IRM-05/IRM-05-spec.pdf.

[7] Texas Instruments, LM7800 Series 3-Terminal Positive Voltage Regulators, Datasheet. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm7800.pdf.

[8] Espressif Systems, ESP32-C3 Wi-Fi and BLE SoC, Datasheet. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf.

[9] D. Power, "How to calculate battery run time," Lithium Ion Battery Manufacturer and Supplier in China-DNK Power. [Online]. Available: https://www.dnkpower.com/how-to-calculate-battery-run-time/.

Appendix A: Requirements & Verification tables

Table 4: Audio Processing Subsystem

Requirement	Verification
Power must be supplied to the microphone of 1.8-3.3V.	1. Connect a multimeter to the power source and the microphone to measure the voltage. 2. Turn on the power source and measure the supplied voltage with the multimeter. 3. Verify that the measured voltage is consistently within the 1.8-3.3V range. 4. Subject the microphone to different operational conditions (e.g., different temperatures, loads) and measure the voltage again.
The microphone must be able to capture audio data within 20 dB of the actual sound to ensure an accurate representation of the environment or video. Without this, the LED response will not be accurate and reflect the real audio.	1. Prepare a calibrated sound meter to generate known sound levels. 2. Produce sounds at different decibel levels (e.g., 60dB, 80dB, 100dB) near the microphone. 3. Measure the output from the microphone using the reference sound meter. 4. Check if the microphone captures audio within ±20dB of the actual sound level generated.

Table 5: Heart Rate Monitor Subsystem

Requirement	Verification
Power has to be supplied to the heart monitor or else the heart monitor would stop working.	1. Connect the heart rate monitor to its power source. 2. Connect the multimeter probes to the power supply terminals of the heart rate monitor. 3. Turn on the power supply and record the voltage across the heart rate monitor's power source. 4. Ensure that the voltage reading is within the acceptable 3.3V input range.

	5. If the voltage is within 3.3V, the test passes. If not, troubleshoot the power delivery system.
The Wi-Fi transmitter also has to work for seamless wireless data transfer to the Wi-Fi receiver in the mounted subsystem, or else we cannot analyze the data to provide a proper output.	1. Ensure the heart rate monitor is transmitting data to the Wi-Fi transmitter. 2. Initiate a data transmission from the heart rate monitor to the Wi-Fi receiver and start the timer. 3. Measure the time it takes for the data to be received on the mounted subsystem. 4. Ensure that the data transmission time is within the 150ms range. 5. If the data transmission occurs within 150ms, the test passes. If it exceeds 150ms, investigate possible transmission delays or interference.
Heart rate monitor must work and not be faulty to ensure that correct data is transferred	1. Use a reference heart rate monitor (e.g., Apple Watch) known for accuracy, and place it alongside the heart rate monitor being tested. 2. Record the heart rate readings from both the reference monitor and the test monitor. 3. Compare the readings to ensure the heart rate monitor being tested is within a 10 bpm accuracy range of the reference monitor. 4. Ensure that the test monitor provides accurate readings across a heart rate range of 60 bpm to 150 bpm. 5. If the heart rate monitor is consistently within 10 bpm across the entire range, the test passes. If not, the heart rate monitor may be faulty or require recalibration.

Table 6: LED Control Subsystem

Requirement	Verification

Must be able to change the LEDS over 20 different colors. If the colors don’t change, the LEDs don’t have any effect on the user or the user's experience.	1. Ensure that the LED strip is powered and connected to the control system. 2. Run a color change cycle to transition through at least 10 different colors. 3. Observe the LED strip and manually verify that the color transitions occur smoothly through at least 10 distinct hues. 4. Continue the color cycle and check that the LED strip can display over 20 different colors as required.
The LEDs must be able to pulse a maximum 120 beats per minute. If the LEDs don’t pulse, then the user won't be able to experience the sync of whatever they are watching to the LEDs	1. Configure the LED control system to pulse the lights at varying speeds up to 120 beats per minute. 2. Position a light sensor to accurately measure the pulse rate of the LEDs. 3. Record the actual pulse rate of the LEDs using the light sensor to ensure it matches the expected rate. 4. Ensure that the measured pulse rate is within a 20 bpm tolerance of the target 120 bpm rate. 5. Visually observe the LED pulse to ensure that the color transition and pulsing effect is smooth and consistent across the range of 120 bpm.

Table 7: Power Management Subsystem

Requirements	Verification
Must not draw more than 500mA of current to preserve battery life.	1. Connect a multimeter to the power source to test how much current is being drawn. 2. Measure the current drawn to the entire subsystem using a multimeter in series with a power supply. Ensure that this current does not exceed 500mA.

If the battery dies, then the fundamental part of the entire system (heart rate monitor) doesn't contribute to the user experience.	1. Monitor battery life before performing any tests and ensure it provides a consistent 3.3V.

Table 8: Communication Subsystem

Requirements	Verification
Must work within a range of 10 meters	1. Set up markers at three different distances from the Wi-Fi receiver. 2. Test within different intervals (2m, 5m, 10m) to ensure that the data is transmitted properly
The Wi-Fi transmitter should successfully pair with the receiver	1. Ensure both the Wi-Fi transmitter (from the wearable system) and receiver (mounted system) are powered on and within the specified range for connectivity. 2. Ensure that any necessary pairing codes or confirmations are provided as required. 3. Observe the connection status on both the transmitter and receiver displays or software interfaces. 4. If disconnections occur, troubleshoot the issue by examining the environment (interference, distance) or the devices themselves.
Must reliably transmit heart rate data from the wearable system to the mounted system without inconsistencies	1. Continuously log heart rate data from the wearable system as it transmits to the mounted system. 2. Compare the heart rate data from both logs to ensure consistency.

Table 9: Computation Subsystem

Requirements	Verification

Must receive and process audio information from the audio processing subsystem	1. Ensure the audio processing subsystem is powered on and connected to the Microcontroller Unit (MCU). 2. Transmit a range of audio signals (e.g., tones or music) from the audio processing subsystem to the MCU. 3. Check the MCU's input to ensure it is receiving the audio signals. 4. Utilize appropriate tools (e.g., oscilloscope or sound level meter) to visualize the received signals.
Must accurately process the audio and heart rate data to convert it into the correct RGB values and luminescence that the LED driver can interpret for color changes	1. Input test data for audio (between 40-60 dB) and heart rate > 100 bpm and ensure that luminescence is bright and color is red 2. Record the RGB values and luminescence levels to verify consistency with the audio and heart rate data.

Appendix B: Circuit Schematics

Mounted System

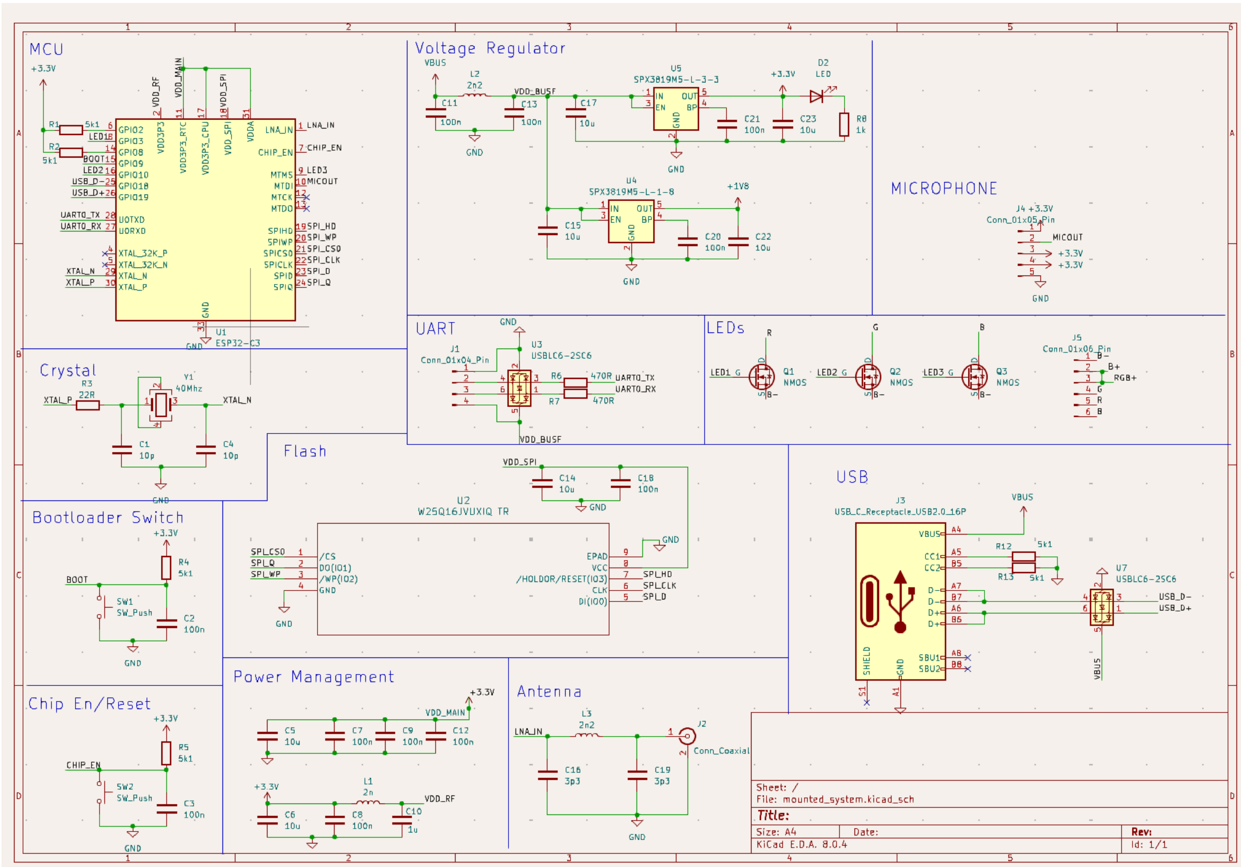

Figure 11: Mounted System Schematic

Wearable System

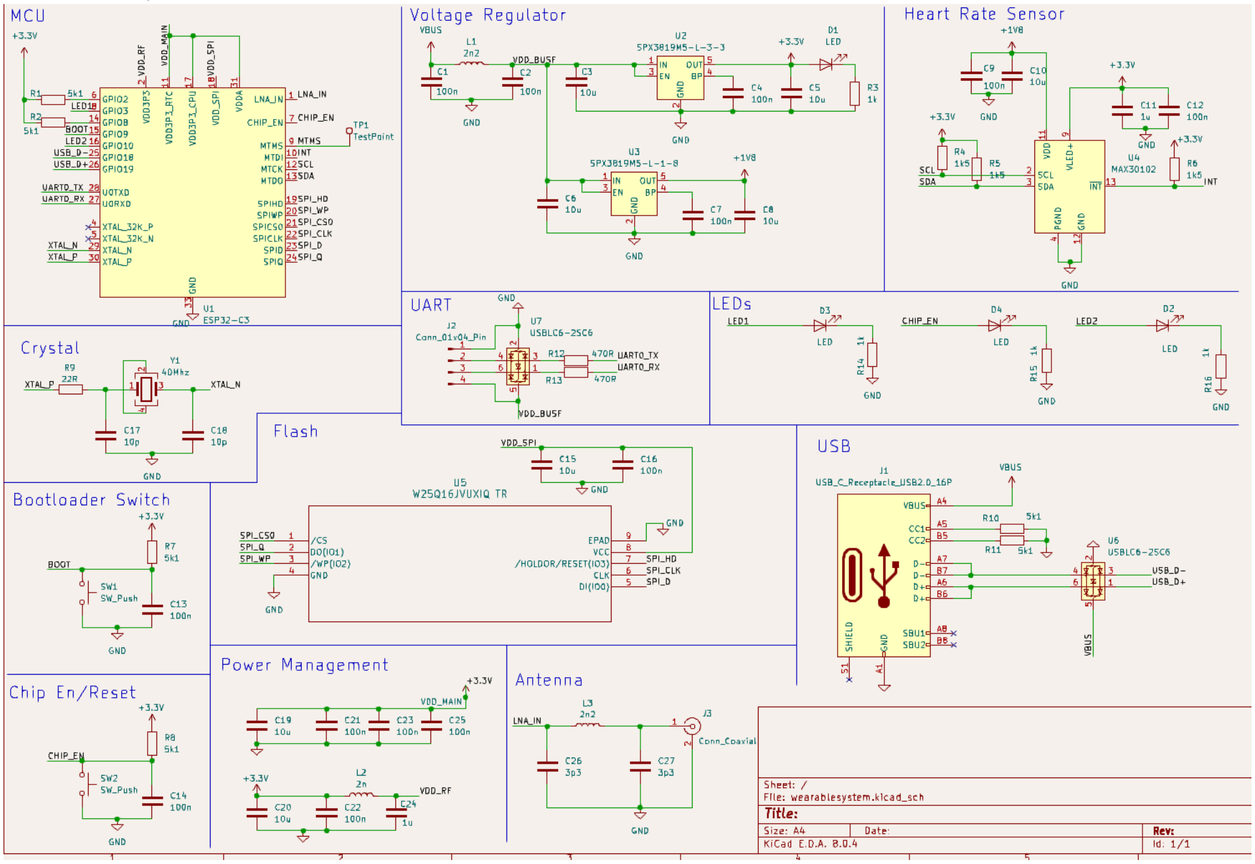

Figure 12: Wearable System Schematic