SURGICAL LED LAMP

By:

Jeremy Wu

Manogna Rajanala

Yogavarshini Velavan

Final Report for ECE 445, Senior Design, Fall 2023

TA: Jason Zhang

6 December 2023

Project No. 36

Abstract

This paper documents the design process, test process, and result of Professor Gruev's pitch project, Surgical Light-Emitting Diode (LED) Lamp. The Surgical LED Lamp aims to aid surgeons in spotting cancerous skin cells and provide better visibility for surgeons during surgery. This document starts with the idea that prompted the project before it goes into the design, explaining the decisions made to build this project. Then, this paper will describe the testing done to verify the project before discussing the future of the project. Ultimately, this project achieved the high-level requirements separately using parts of the Printed Circuit Board (PCB) with a development board. Our project is implementing a surgical LED lamp that will accompany a medical microscopic camera; medical professionals can locate malignant cells during surgery and remove cancerous growth. Our device consists of two variants of LED lights: white light LED and infrared LED.

Contents

1. Introduction

1.1 Problem

In surgical settings, medical professionals often have to employ different methods to detect and remove cancerous or malignant cells. This process is gruesome and requires a deep analysis of areas affected by the cancerous cells; due to the intricate nature of the human body and tissues affected by malignant cells, it can be challenging to remove affected areas accurately. While medical devices can assist in eliminating cancerous cells, it is often the human vision that restricts and limits surgeons when trying to locate and remove cancerous cells. There needs to be a device that allows surgeons to expand their eyes to detect more significant amounts of growth and early, small areas that are affected by the cancerous cells. This device would assist in making incisions in the right place.

1.2 Solution and Visual Aid

1.2.1 Solution

A device can mitigate this issue by expanding human vision and allowing individuals to see more clearly in surgical settings. Our solution that aligns with this goal is a programmable surgical light that works together with a microscopic camera. Research led by the Institute of Cancer Research states the usefulness of infrared light to detect cancerous cells; infrared light can force cancerous cells to illuminate under the light, giving surgeons a much clearer understanding of where affected areas are [1]. Cancer being such a leading cause of death, it would be crucial to be able to detect cancer early to allow early removal of malignant cells. Our solution consists of three PCB, components of each PCB will be discussed later in this report.

1.2.2 Visual Aid

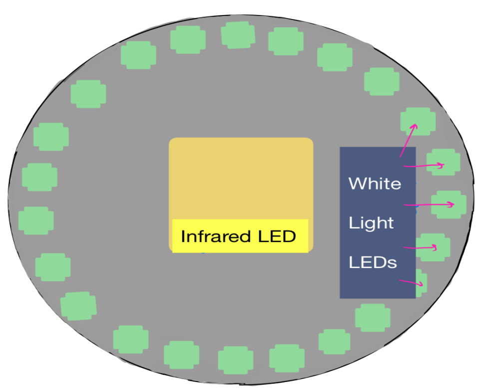


Figure 1. Visual Aid

1.3 High Level-Requirements and Functionality

In a broad sense, our project had three main objectives that it had to fulfill to be considered successful:

1. The twenty-four white light LEDs(light emitting diode) must light up to provide enough illumination for the surgeon to see the affected area.
2. We should be able to light the infrared LED light so the surgeon can use an infrared camera to pick up on this light.
3. Finally, a mechanism should be able to adjust the LEDs' brightness based on the user's preference.

The infrared and white light LEDs are placed on a PCB on its own so that users may mount it to illuminate the areas they are working on. There are three switches that the user can control. The first switch is a master switch that turns both types of LEDs ON or OFF simultaneously. The second switch is for the white light LEDs which would turn OFF or ON the set of twenty-four white light LEDs without affecting the functionality of infrared LED. The final switch controls the infrared LED independently of the white light LED. In addition to the switches, two potentiometers control the brightness of the LEDs. If the LEDs are ON, the potentiometer can be used to independently increase or decrease the brightness of each type of LED.

1.4 Subsystem Overview

Our project consists of four main subsystems: user-interface, control, power supply and voltage regulator, LED drivers and LEDs.

1.4.1 Block Diagram

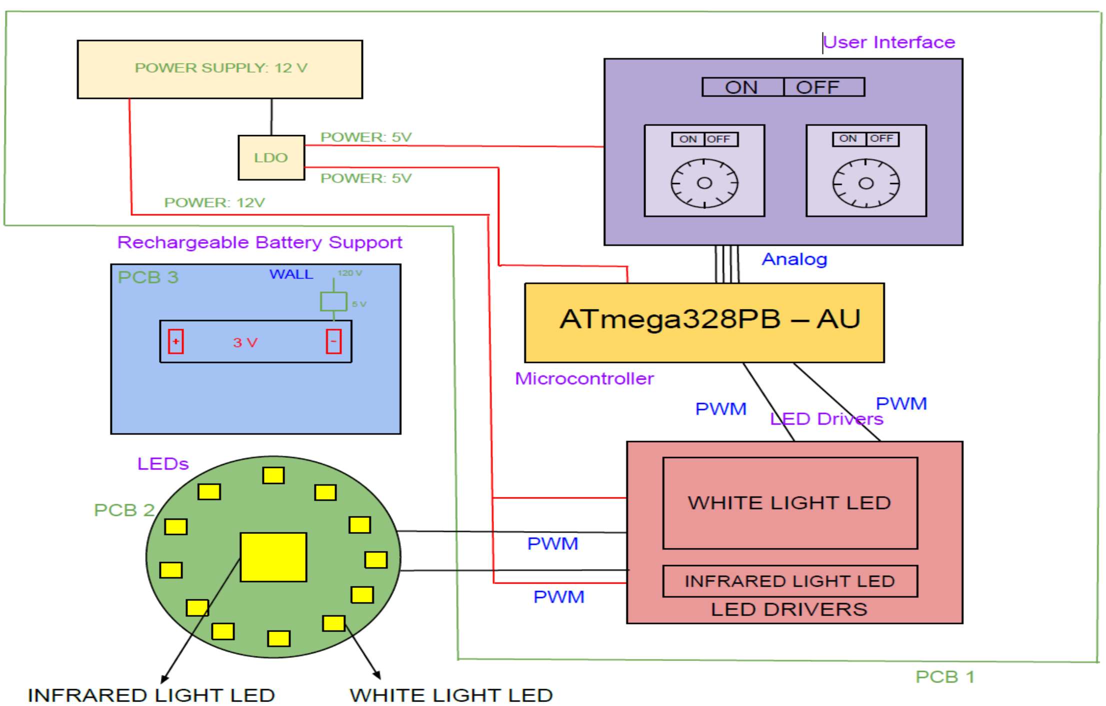


Figure 2. Block Diagram

1.4.2 User Interface Subsystem

The user interface subsystem allows the user to control aspects of the surgical light. This subsystem sends signals to the microcontroller as analog and digital input pins. The microcontroller then uses these values to create a Pulse Width Modulation (PWM) output to send to the LED drivers and allow the user to adjust the brightness of the LEDs.

1.4.3 Microcontroller Subsystem

The microcontroller subsystem interprets values from the user interface into PWM signals. The microcontroller will read the switch values from digital pins and potentiometer values from analog pins. Based on those values, the microcontroller produces PWM signals that will turn on or off or adjust the brightness of our LEDs. There will be two PWM signals, one for each kind of LED. For our microcontroller, we are using the ATmega328PB-AU[3].

1.4.4 Power Supply Subsystem

The power supply subsystem provides power to all of our subsystems. The power is provided by a series of three 18650 rechargeable batteries, which will supply a maximum of 12.6 V. From the power supply, we utilize different voltage regulators to step down the voltage for each subsystem. We had various LDO(Linear Dropout Regulators), LED driver boost and buck converters to decrease our voltage, as 12.6V is more than most subsystems require.

1.4.5 LED Driver Subsystem

To drive the LEDs, we required LED drivers that would be able to change the brightness of the LEDs by modifying the current that is being sent to the LEDs through PWM dimming technique. The project required two types of LED drivers: white light LED driver and infrared LED driver since each of the LEDs required different voltage requirements. The white light LED driver, ISL97682IRTZ-TK would boost the 12.6V input voltage to 36V for the two sets of twelve white light LEDs and the infrared LED driver, TPS54200 would buck the 12.6 input voltage to 2.8V

1.4.6 LEDs

The LED subsystem consists of twenty-four white light LEDs and one infrared LED. The white light LEDs have a nominal voltage drop of 3V and a nominal current requirement of 60mA. The white light LEDs have to be bright enough to help the user see the affected area. The component that we used was MP-2016-1100-27-90 warm white light LED[4]. For the infrared light LED, we had a power requirement of 1W. The component that we chose for the infrared LED was the SMD3030 1W IR 780 nm High Power LED[5].

2 Design

2.1 User Interface

The user interface was the bridge connecting the user to the microcontroller and ensuring that the user could fully control the brightness of each type of LED system.

2.1.1 Design Description & Justification

The user interface comprises three toggle switches and two 100k ohm potentiometers. The three SPST(Single-pole single-throw)switches determine the states of the LED system, where the first switch is the main switch that turns on the entire LED system. An SPST switch is used due to the sufficiency of having an on/off functionality for the switches. The toggle switch has a minimum voltage requirement of 20mV and a max rating of 28V. The acceptable range for current through the switches is around 0.1mA to 0.1A.

The two potentiometers are used to manage the brightness of the two sets of LEDs. Each of the 100k ohm potentiometers sends the appropriate voltage to the microcontroller, allowing the microcontroller to create a PWM output that is sent to the LED drivers. To allow for a wide range of resistance values from the potentiometer, we employed 100k ohm potentiometers with a knob to allow easy modification of the potentiometer values. The variable resistance of 100k ohm enables the potentiometer to modify the brightness of the LED systems with finer control.

2.1.2 Design Alternatives

The SPST toggle switch was used instead of an SPDT(Single-Pole Double-Throw) switch because the SPST toggle switch is more cost-effective when the requirements do not include switching between multiple circuits or positions. In our design, the switch must be in one of two positions; therefore, the SPST switch is enough. The voltage rating also allows the switch to take 5V, the amount of voltage we provide the user interface subsystem with. The usage of a 100k ohm potentiometer compared to a 10k ohm potentiometer was considered; the 100k ohm potentiometer was the better choice due to the ability to have better and finer control over the LED brightness due to the more extensive range of values that can be allowed with the 100k; this allows for a more precise and detailed setting for the brightness for the LEDs.

2.2 Microcontroller

The microcontroller was responsible for interpreting our user interface values and generating PWM signals to control the brightness of the LEDs. The microcontroller sent out two different PWM signals to control each kind of LED separately.

2.2.1 Design Description & Justification

The microcontroller we used for our project was the ATmega328PB-AU. We first decided to use the ATmega328 because of our familiarity with the Arduino Uno. Since the Arduino Uno uses the

ATmega328, we wanted to stick to something comfortable. Then, we decided to use the ATmega328PB-AU instead of other versions like the Atmega328P, which the Arduino Uno uses, because of its ability to produce two 16-bit timers[6]. The 16-bit timers would allow us to output two PWM signals ranging from 0 to 65535, much bigger than the 8-bit timers, which only range from 0 to 255. By using the 16-bit timers, we would have finer precision over the LED's brightness, giving the user more control over the desired brightness.

2.2.2 Design Alternatives

An alternative we should have considered was the ATmega328P. Although having the two 16-bit would be ideal for the user to have precise control over the brightness, there was an issue when we tried to compile our code. When we were trying to write to specific output registers that we found based on the datasheet, the library we used was claiming that they did not exist. Since Arduino does not use the ATmega328PB or ATmega328PB-AU, we had to find our library online to install it. However, they wouldn't compile with the different libraries as they all had specific requirements, like clock and clock frequencies, that didn't match our setup. As a result, we could not find a compatible online library to compile our code. Since the Arduino Uno uses the ATmega328P, they would already have the correct library installed, avoiding our issue with the ATmega328PB-AU. Even though we would have to use the 8-bit timers, they would still be sufficient to get the project to behave as intended.

2.3 Power Supply and Voltage Regulator

The power supply was a series of three 18650 rechargeable batteries with a total output of 12.6V. These batteries would power the other subsystems and the appropriate regulators to decrease the voltage.

2.3.1 Design Description & Justification

We decided to use the three 18650 rechargeable batteries for the power supply because of our LED boost driver. Since we have twelve white LEDs per group, each has a 3V drop, we would need 36V to power all 12 LEDs. As a result, the LED boost driver would boost the power supply up to 36V. Having to provide 36V would imply that the duty cycle the boost has to generate would be:

$$Vout \ = \ Output\,Voltage, \ Vin \ = \ Input\,Voltage, \ D \ = \ Duty\,Cycle$$

$$Vout \ = \ Vin\,/\,(1 \ - \ D) \ -> \ 36V \ = \ 12.6V/(1 \ - \ D)$$

$$D \ = \ 0.65$$

Having the duty cycle at 0.65 is reasonable. However, if the input voltage were any lower, like using two batteries, the duty cycle would be much more significant, which makes it harder for the LED boost to generate that kind of duty cycle. If the voltage were any higher, the other regulators would have a hard time dropping down that much voltage. The regulators already have to drop 7.6 volts, and any more significant input voltage would cause the regulators to have a greater chance of failing.

Additionally, the other subsystems of our project only need 5Vs. Since power supplies are 12.6V, we need to use LDO(Linear Dropout Regulators) to decrease the voltage from 12.6V to 5V. We used a TQL851CSV50 RLG, which is a fixed 5V Output LDO. This LDO could take 3.3V to 50V and output 5Vs, powering our microcontroller and user interface [7].

2.3.2 Design Alternatives

An alternative we could have used instead of the LDO is a buck converter to step down the voltage. Having to drop 7.6V across the LDO constantly would burn it out quickly as LDO deals with excess voltage through heat. A buck converter is more efficient at stepping down voltage, which means that it dissipates less heat when stepping down voltage. As a result, the buck converter would be a safer option to step down significant voltages.

2.4 LED Drivers

2.4.1 Design Description & Justification

The LED driver subsystem comprises two LED drivers: a white light LED driver and an infrared light LED driver. These LED drivers take the PWM signal produced by the microcontroller as the input and then make a PWM signal with the same duty ratio but at a higher frequency and amplitude and send this to the LEDs. We used LED drivers with PWM signal capabilities since the PWM dimming method is much more accurate than the conventional analog dimming technique[8].

For the white light LED driver, we used the ISL97682IRTZ-TK component. We decided to use a boost LED driver instead of having a separate boost converter before the LED driver since we tried to minimize the number of power converters in our design for efficiency considerations. Additionally, we chose this specific component for our boost LED driver as it had two outputs that could drive the two strings of LEDs. The LED driver produces PWM signals at a fixed frequency of 30kHz, much higher than frequencies the human eye can detect to eliminate flickering observations. It comprises eight capacitors, seven resistors, a Schottky diode, and an inductor. The maximum output current that the LED driver could deliver to the white LEDs was set by the value of the resistor, Rset:

$$Rset = 804/Peak\ LED\ Current = 804/0.06A = 13.8k\ Ohms$$

We also used an inductor with inductance of 40uH to constrain the output current ripple. The formula used to calculate the inductance is in Appendix D.

We utilized the TPS54200 synchronous buck converter LED driver for the infrared LED driver since the infrared LED required 2.8V, and the input to the LED driver was the primary battery source of 12.6V. We chose this specific component since it was relatively simple to implement and had an extensive and detailed datasheet. We designed this LED driver to produce a PWM signal at a frequency of 400kHz, which is much higher than the white light LED driver, as the infrared LED light can be observed through a specialized microscopic surgical camera and not the human eye. The design for this converter consists of five capacitors, three resistors, and an inductor.

The maximum output current that the LED driver could deliver to the infrared LED was set by the value of the resistor, Rsense:

$$Rsense \ = \ 1OOmV/Peak\ LED\ Current \ = \ 100mV/350mA \ = \ 0.285\ Ohms$$

We also used an inductor with inductance of 40uH to constrain the output current ripple.

2.4.2 Design Alternatives

Initially, we considered using a buck and boost converter separately from the LED drivers. Although this would have reduced the complexity of the LED drivers, it would also mean that we would be incurring more significant power losses since the current has to pass through two DC-DC converters instead of one. During our final week, we also realized that it would have been better for us to use two white LED drivers instead of one dual-output white LED driver. Having two LED drivers would have divided the power required for the white LEDs between two LED drivers, minimizing the risk of component failure.

2.5 LEDs

2.5.1 Design Description & Justification

We employed two kinds of LEDs for our LED PCB.

The first kind of LED, the white light LED, was used to illuminate the area being worked on. Therefore, we wanted the white LED system to have a moderately high power rating. We used the MP-2016-1100-27-90 warm white light LED with a power rating of 0.18W. Therefore, our total white light LED power delivered was:

$$P \ = \ 24 * 0.18W \ = \ 4.32W$$

The total power output of 4.32W is enough to provide reasonably good radiance while at the same time ensuring that the LEDs do not overheat on the PCB. The current rating of these LEDs was 60mA, therefore we did not have to increase the track widths associated with the LEDs and the LED drivers on the PCBs.

The second type of LEDs that we used was the infrared LED. We were required by the specifications of the pitched project to utilize a one-Watt 780 nm infrared LED. Since the choice in the market for LEDs with this exact specifications were considerably few, we chose the Lumixstar SMD3030 1W IR 780nm High Power LED which had a current rating of 350 mA and 2.8V.

2.5.2 Design Alternatives

The choices for the white LED were surplus since we had generic and conventional requirements from the white LED. It is likely that any of the other reasonably similar LED components would have worked in the design.

As for the infrared LED, suppliers for LEDs with our exact specifications were scarce so that narrowed our choice. However, during demonstration and testing since we had trouble procuring the infrared LED, we chose a blue LED in the visible spectrum range with the exact voltage and current requirements as the infrared LED to demonstrate the working of the infrared LED driver and design. As an additional aspect, testing with a blue LED greatly reduced the risk of accidental damage to the eyes of the testers.

2.6 PCB

Our project consisted of three PCBs, one for the microcontroller, one for the LEDs, and one for a rechargeable battery circuit.

2.6.1 Design Description & Justification

The PCB design for our microcontroller consisted of a couple of different rounds of modifications. The main design choices when working on this PCB were to make sure that each subsystem had components that were close to each other, specifically the capacitors, to make sure that the components that used the capacitors were stable and were not seeing any sudden spikes in voltages which could cause an overvoltage and ultimately failure of the component. Moreover, we tried to have some physical space between the subsystems. This ensured the design was modular and clear during the soldering and testing process. This also safeguarded the other subsystems in case a component blew up.

The primary consideration during the design of our LED PCB was reducing the amount of heat accumulation on the PCB. The LEDs themselves were placed as far apart as possible so each LED had as much contact with open air as possible. Additionally, we designed the PCB on a single layer of copper. This was to ensure that the PCB board has faster heat dissipation when in contact with air.

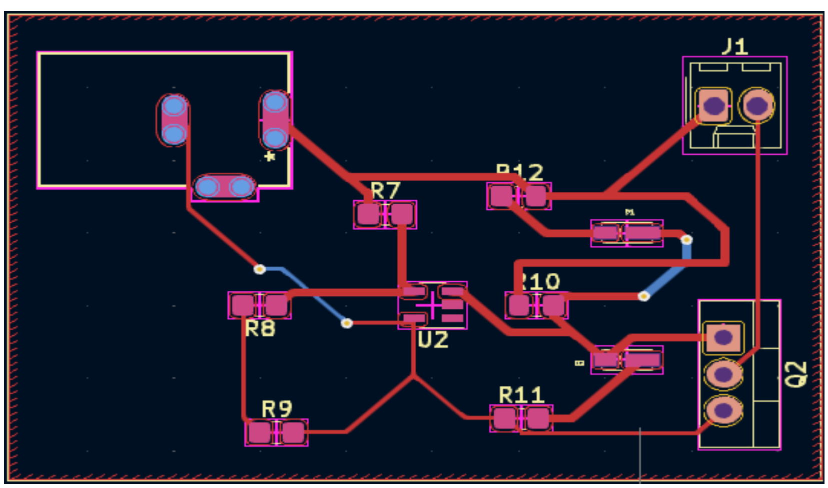
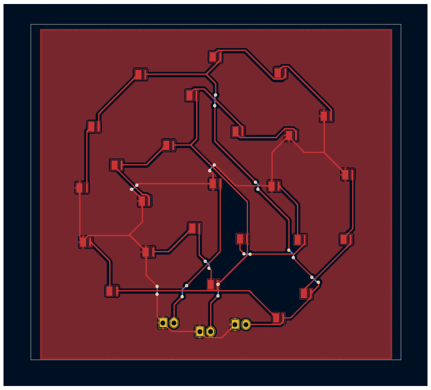


Figure 3. Rechargeable Battery PCB



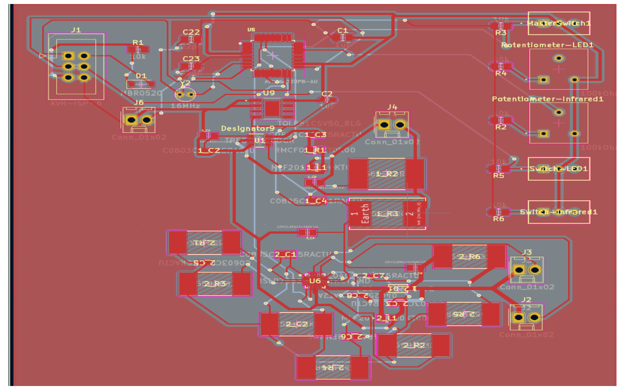


Figure 4. LED PCB

Figure 5. Microcontroller PCB

2.6.2 Design Alternatives

As an alternative design, the microcontroller PCB would have been a little bit smaller, since the components fit on the PCB with a vast amount of free space. Furthermore, we would size our resistors to be a little bit smaller in order to allow for a smaller PCB that could be used as a handheld design. Lastly, we would have added more filtering components as we would want to decrease the noise created by the components such as the voltage regulators.

As an alternative, the LED PCB could have used a Metal Core PCB instead of the standard FR-4 material which would have better heat transfer properties and are conventionally used during LED PCB design. However, we had to forgo this option due to budgetary constraints. Instead, we mitigated the issue of heat dissipation by designing our LED PCB to consist of a single layer of copper so heat would not be trapped between the layers.

Alternatively, the rechargeable PCB could have used a regulator or buck converter after the DC jack to allow a DC power adapter. Since not everyone has a 5V DC power adapter, the buck converter would relieve the need to use a 5V, and the users could use different voltage DC power adapters and still be able to charge their batteries.

2.7 Software

The software for the microcontroller reads the input digital and analog pins before converting them into a PWM output.

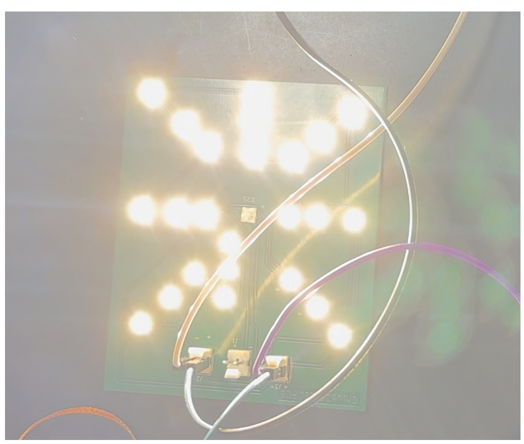
2.7.1 Design Description & Justification

The software would set up the correct PWM by writing to TCCR0A and TCCR0B before doing anything. Then, it would read from each pin and use the read values to determine the duty cycle[9]. Since we used two 8-bit timers and the input was a 10-bit input, we had to convert the input to a 0 to 255 scale before we could write it to the output register. Additionally, there were instances where all of the pins would read 0 for a quick instance. To mitigate this, we added a 100ms delay whenever we saw a change in the input values before we reread the input values. The delay would allow the pins to read the original values if the previous values read weren't an accident.

2.7.2 Design Alternatives

An alternative to the software design would be to have ranges of values linked to a particular output value. For example, if the analog reads anything from 0-20, the output should be 0. This idea would be better as the light wouldn't change at every instance the read value is changed. Something accidentally touches a potentiometer, like a gust of wind, which would change the value. This range of values linked to an output value would give a buffer to prevent slight accidents that change the output.

Figure 6. Blue LED



` **Figure 7. White LEDs**

3. Requirements and Verifications

3.1 Power Supply and Voltage Regulators

3.1.1 Power Supply

We had a requirement that the three 18650 batteries produced a voltage of 12V +/-0.6V for at least four hours to power the other subsystems. We would measure this using a voltmeter at the input power pins of our main PCB. Our batteries were able to output voltages within this constraint. 1However, since our linear-dropout voltage regulator failed during modular testing, we could not use our batteries with the PCB as it would have damaged the rest of the system.

3.1.2 Voltage Regulators

The requirement for the voltage regulator was to be able to output 5V and maintain this output for an extended period. We would measure the output of the LDO using the voltmeter when we first connected the power supply to the PCB. However, when we first tested it, we realized the output was a 3V drop from our input, even though the documentation said it was a fixed 5V. Eventually, the LDO would not drop any voltage, no matter what we gave it since it short-circuited when we increased the voltage above 10V.

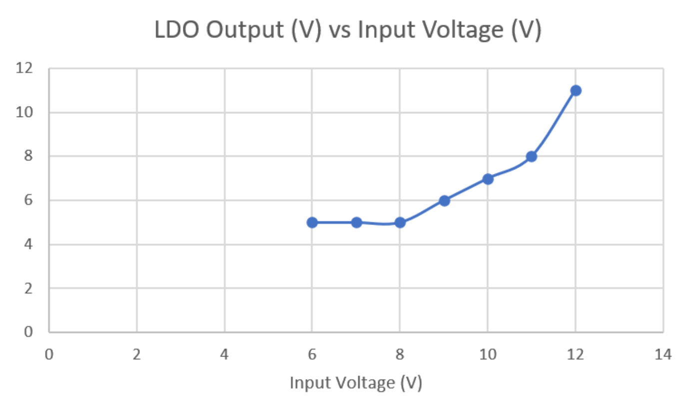


Figure 8. LDO vs Input Voltage

3.2 User Interface and Microcontroller

3.2.1 User Interface

The requirements for the user interface mainly included the ability to employ switches and potentiometers to control the brightness of the LEDs. The switches worked as expected; the voltage was checked across the switch when turned on and off and correctly turned on the respective LED system using a voltmeter. The potentiometers also worked successfully, as we could send the potentiometer's voltage and create a PWM duty cycle as an output to set the brightness of the two LED systems. This was verified using the knob of the potentiometer and making sure that the brightness of the LEDs increased or decreased respectively.

3.2.2 Microcontroller

The main requirement for the microcontroller was to have the functionality of taking in five different signals from the user input (analog and digital) and outputting a PWM duty cycle that would determine the brightness of the appropriate LED system. The PWM would be sent over to the LED drivers. We could not verify this with our microcontroller due to a short circuit in our PCB; however, we successfully tested this with our Arduino dev board, as seen below.

```
14:21:17.202 ->  Master:1 WhiteSwitch:1 InfraSwitch0 WhiteAnalog:623 InfraAnalog:0
14:21:17.267 ->  Master:1 WhiteSwitch:1 InfraSwitch0 WhiteAnalog:623 InfraAnalog:0
14:21:17.332 ->  Master:1 WhiteSwitch:1 InfraSwitch0 WhiteAnalog:623 InfraAnalog:0
14:21:17.396 ->  Master:1 WhiteSwitch:1 InfraSwitch1 WhiteAnalog:623 InfraAnalog:80
14:21:17.461 ->  Master:1 WhiteSwitch:1 InfraSwitch1 WhiteAnalog:623 InfraAnalog:80
14:21:17.556 ->  Master:1 WhiteSwitch:1 InfraSwitch1 WhiteAnalog:623 InfraAnalog:80
```

Figure 9. Switches and PWM Output Signals

3.3 LED Drivers and LEDs

3.3.1 LED Drivers

The main requirements of our project lie in the LED drivers being able to produce a PWM signal with the correct duty ratio as requested by the microcontroller.

Change in Duty Ratio As the user tunes the knob in the user interface, the microcontroller produces different duty ratios, and the PWM signal produced by the LED driver should also increase or decrease accordingly. We tested our knob at a duty ratio of 3% and a duty ratio of 90% and the intermediate values. The output current of the infrared LED driver increased to match it as tested on an oscilloscope. There was an average error of about 3% in the duty ratio. We were unable to get the white LED driver working. It did not produce an output PWM signal. This could be a component failure during the soldering process.

Maximum Output Voltage The infrared LED driver is supposed to produce 2.8V at a nominal current of 350mA. When measured with a voltmeter, the output voltage was 2.98V, and the current was 377mA. Since this was within the tolerable limits of the LED, the bucking action of the driver was successful. On the other hand, the white LED driver produced about 2V of output and did not light the white LED. We presumed that this driver failed due to overheating.

3.3.2 LEDs

Brightness of the White LED We had a qualitative requirement of ensuring that the white LEDs were bright enough to illuminate the area of the skin. When we supplied the white LEDs with an external power supply of 48V and 120mA, it was extremely bright, as seen below, almost blinding. Figure 7. contains the result of the test.

Wavelength of the Infrared LED We had a strict requirement that the infrared LED produce light from 780nm to 785 nm. We would measure the wavelength produced using a spectrophotometer. However, we could not elicit a response from the supplier for the infrared LED, so we had to use a replacement blue-light LED with the same voltage and current requirements.

4. Costs

4.1 Parts

Table 1. Parts Costs

Component	Part Number	Manufacturer	Quantity	Unit Cost($)	Actual Cost ($)
PCBs		PCBWay	5	25	25
White LED Driver	ISL97682IRTZ-TK	Renesas/ Intersil	4	3.40	13.92
Infrared LED Driver	TPS54200DDCR	Texas Instruments	4	1.47	5.88
White Light LEDs	MP-2016-1100-27-90	Luminus Devices Inc.	26	0.096	2.496
Capacitor 0.1uF	C0603C104K5RAC7867	Kemet	10	0.074	0.74
Capacitors 33pF	C0805C330K5RAC7800	Kemet	5	0.24	1.20
Inductor 39uH	MLF2012K390KT000	TDK Corporation	10	0.14	1.40
Resistors 10kOhm	355010KFT	TE Connectivity	40	0.01	0.40
Blue Light LED (Infrared LED replacement)	SST-10-SB-B130-M470	Luminus Devices Inc.	3	1.83	5.49
Microcontroller	ATMEGA328PB-AU	Microchip Technology	1	1.81	1.81
Crystal 16MHz	520-HCA1600-SX	ECS	2	1.04	2.08
Diode MBR0520	MBR0520L	Onsemi	4	0.45	1.80
AVR-ISP-6	61300621121	Würth Elektronik	6	30.57	3.42
Switches	360-2973-ND	NKK Switches	3	4.43	13.29
Potentiometers	3310Y-002-104L-ND	Bourns Inc.	2	3.45	6.90
Rechargable batteries	18650	Samsung Electronics	4	6.99	27.96
Battery holder		Hiecube Electronics	1	12.99	12.99

Miscellaneous					20.00
Total					146.776

4.2 Labor

The average salary for a graduate with a bachelor's degree in ECE is $87,769 (Electrical Engineering) and $109,176 (Computer Engineering)[10]. Taking an average of this lends to an average salary of $98,472 yearly.

If we consider 40 hour work weeks, that would amount to $47.32/hour. On average, we spend about 20 hours each week, over a span of ten weeks this semester.

The total labor cost is as follows:

$$Labor\ Cost/Person\ = \$47.32/hour\ *\ 20\ hours/week\ *\ 10\ weeks\ =\ \$9,464/\ person$$

$$Total\ Labor\ Cost\ = Labor\ Cost/Person\ *\ 3\ people\ = \$28,392$$

4.3 Total

$$Parts\ Cost\ =\ \$146.776$$

$$Labor\ Cost\ =\ \$28,392$$

$$Total\ Cost\ =\ \$28,538.776$$

5. Conclusion

5.1 Accomplishments

Our project was able to turn on our two kinds of LEDs. Both sets of twelve white LEDs can turn on when passing 36V and 0.12A through them. Additionally, our infrared LED, which was replaced with a blue LED with the exact specifications due to the lack of cooperation from the infrared LED seller, can turn on. Finally, we can produce accurate PWM signals from the microcontroller after receiving different voltage values from our switches and potentiometers. With these PWM signals, we can control the LED's brightness.

5.2 Uncertainties

We believe that one of the main reasons our project was not fully functional was due to the short circuit that we faced. Our LDO short circuited; due to time constraints, we were unable to replace that component with a Buck converter and order another round of a PCB design. This short circuit also made us incapable of testing with our microcontroller. We believe that the short circuit might have occurred due to either the overheating of the LDO while using a heat gun or the constant 7 volt drop across the regulator. Furthermore, our microcontroller did not function as desired; we determined that this was due to the lack of compatibility between the bootloader and the software that we were trying to program the microcontroller with. We also believe that there could have been a difference in the clock speed of the bootloader and the microcontroller even though we were attempting to set the clock speed to be consistent between the two. Lastly, our white LED driver did not function as expected; we believe that this was due to the overheating of the component when using a solder gun or the short circuit which caused the driver to stop working appropriately. With these setbacks, we were unable to get our project to be fully functional; however, we were able to achieve most of our goals that we set up for this project.

5.3 Ethical considerations

While working on this project, ensuring safety and ethics were considered when designing and testing different components was essential. One of the main concerns was ensuring that while testing the LEDs, we were taking care of the users of the LED PCB. This is due to the brightness of the LED systems that can cause potential damage to the eyes. The first code in the IEEE Code of Ethics states “the safety, health, and welfare of the public.” This code of ethics was employed by making sure to be aware of when the LEDs’ brightness would be turned up while testing the LED subsystem. While we could not test the device with patients, we would consider the implications of having the infrared light affecting a user’s eyes and skin in the case where the device would be used.Furthermore, we would ensure that users are fully aware of the lasting impacts or effects of the infrared and white LEDs. While this device may aid in detecting cancerous cells, it should be made clear that misdiagnosis may occur due to the complex nature of tissues and malignant cells. OSHA(Occupational Safety and Hazard Administration) states that “workers need to wear protective, electrical personal equipment to ensure that they are safe and not being harmed”[2]. We made sure to be aware of when the LEDs were being tested, and the brightness was set not to hurt our eyes as we tried the LED system.

5.4 Future work

Currently, the User Interface subsystem contains three switches and two potentiometers, which are soldered onto the PCB. In the future, we want to utilize a digital platform, either a mobile application or a web application. A digital platform will give users more control and a better experience with the controls. We would have to use a different microcontroller, like the ESP, to allow the PCB to access wifi or Bluetooth. This would allow for communication between the PCB and the application. Additionally, we want to add motion sensors to our projects for a better understanding when performing surgery. Since surgeons' hands will be preoccupied with performing surgery, having sensors would allow them to control the lights without the risk of touching anything that hasn't been disinfected. Adding hand sensors would be something that would be added on top of the design, as having the switches and potentiometer are still helpful. Lastly, we want our project to get incorporated with the microscopic camera to utilize the infrared light. Another design change that we could incorporate would be to have the light be a handheld rather than the overhead device that we have worked on.

5.5 Acknowledgements

As we approach the conclusion of working on this project, we would like to thank all the faculty and staff that have assisted us in being successful to the extent that we were capable. While we were not able to get our project fully functional, we are very thankful to have gotten the experience of working on the project due to the lessons and knowledge gained throughout the duration of this project. Specifically, we would like to thank Professor Fliflet, our TA Jason Zhang, head TA Jason Paximadas, and Professor Gruev in encouraging us and providing us with feedback and insight into how to better our project.

References

[1] “Scientists harness light therapy to target and kill cancer cells in world first,” The Guardian, https://www.theguardian.com/society/2022/jun/17/scientists-harness-light-therapy-to-target-and-kill-cancer-cells-in-world-first.

[2] “1910.137 - Electrical Protective Equipment. | Occupational Safety and Health Administration,”www.osha.gov.https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.137

[3]“ATmega328PB ATmega328PB Datasheet.” Available: https://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf

[4] MP-2016-1100-22-90 Luminus Devices Inc. - DigiKey. Available at: https://www.digikey.com/en/products/detail/luminus-devices-inc/MP-2016-1100-22-90/8682181.

[5]“SMD 3030 1W & 3W IR 780nm LED - 780nm LED - Lumixtar,” www.lumixtar.com. https://www.lumixtar.com/smd3030-1w-ir-780nm-led.html

[6]“ATmega328PB ATmega328PB Datasheet.” Available: https://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf

[7] TQL851CSV50 RLG Taiwan Semiconductor Corporation | Integrated ... - digikey, https://www.digikey.com/en/products/detail/taiwan-semiconductor-corporation/TQL851CSV50-RLG/16610554

[8]“Application Note A031e PWM vs Analog Dimming of LEDs Pulse Width Modulation (PWM) vs. Analog Dimming of LEDs.” Accessed: Dec. 07, 2023. [Online]. Available: https://www.aimtec.com/site/Aimtec/files/documents/ApplicationNotes/a031e%20pwm%20vs%20analog%20dimming%20of%20leds.pdf

[9]“ATmega328PB ATmega328PB Datasheet.” Available: https://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf

[10]G. E. O. of M. and Communications, “Salary Averages,” ece.illinois.edu. https://ece.illinois.edu/admissions/why-ece/salary-averages

Appendix A Abbreviations

Table 2. Abbreviations

LED	Light Emitting Diode
PWM	Pulse Width Modulation
LDO	Linear Dropout Regulator
PCB	Printed Circuit Board

Appendix B Requirement and Verification Table

Table 3. System Requirements and Verifications

Subsystem	Requirements	Verification	Verification Status (N or Y)
Power Supply	1. The power supply for the microcontroller should be able to continuously supply 12V +/-0.6 V after the batteries are charged by the battery rechargeable circuit.	1. We can check using a voltmeter that there is only 12.6V from the battery.	Y
	2. The rechargeable battery circuit should charge each battery cell to 3V +/-0.2V.	We will use a depleted cell and then see if the battery recharging circuit is then charging it to 3V based on voltmeter reading.	N
Voltage Regulators	1. The LDO is able to supply 5V+/- 0.2V for any power supply from the battery above 6V.	1. We will use a voltmeter to measure the output of the LDO for varying input voltages to the LDO from 6V to 12.6V which is the maximum of the battery.	N
	2. The LDO should be able to continuously supply 5V +/- 0.2V for long periods of time like 30 mins.	We will run our circuit for about an hour and see if the LDO is able to continuously supply the required voltage for longer periods of time without short-circuiting.	N

User Interface	1. The interface should allow user to turn on entire LED system with the main switch	1. The voltage that is being supplied to the switches can be tested to see if the on/off functionality of the switch is correctly changing the voltage. We will be supplying the switches with 5V so the switches will be using a voltmeter to check that the voltage being supplied becomes 0. (Applicable Range for the current will need to be 0.1mA ~ 0.1A at a minimum of 20mV ~ a max of 28V)	Y
	2. Each of the LED systems, infrared and visible, will have a switch to turn on that specific LED system	2. Same as above. The voltage will be checked across resistors for the two LED switches to make sure that the correct voltage is flowing through when the switch for an LED is turned on/off.	Y
	3. The three toggle switches that will turn on the LED system need to be within the max and min voltage ratings so that the LEDs turn on and off when the switch sends the min/max voltage to the microcontroller and LED drivers	3. Check the voltage across the resistor when switch turned off to see the maximum voltage and when the switch is on to see the minimum req so that it is within the bounds of the toggle switches voltage rating.	Y
	4. Once the switches are turned on, the potentiometer can be used to change the brightness of the specific LED.	4. The potentiometer can be tested by having someone adjust the potentiometer while checking the ends of the potentiometer to see if the	Y

		resistance is being adjusted. The potentiometer is 100 ohms so we will make sure that the range of the resistance will be able to reach that by checking the voltage of the resistors with a voltmeter.	
	5. Each of the five components have information that will be sent to the microcontroller through signals that will allow the microcontroller to determine which set of LEDs to turn on/off. This will need to be communicated to the microcontroller using analog and I/O input to the microcontroller.	5. We can use a voltmeter to test the voltage that is being sent to the microcontroller at the analog or I/O pin to make sure that the communication protocol is correctly passing information from the switches and the potentiometers.	Y
Microcontrol -ller	1. The microcontroller should firstly be able to read the potentiometer voltages accurately and if the switches are on or off.	1. We will be checking firstly to see if the signals from the user interface are being sent and correctly interpreted by the microcontroller. We can do this using turning the switches on and off and checking the voltage across the resistors and the resistance for the potentiometer and checking the voltage at the analog and I/O pins to make sure that the values are consistent.	Y

	2. It should be able to produce accurate duty ratio PWM signals to the LED drivers for the two types of LED drivers. The duty ratio is proportional to the voltage across the potentiometer or completely pulled to low if the respective switch or the master switch is turned off. The 5V is around the nominal voltage expected across the potentiometer at its highest resistance setting. $\mathbf{DutyRatio_{PWM} = V_{potentiometer}/5V}$	2. The voltages from the potentiometer are going to be checked by checking the brightness of the LEDs that are turned on. We will be checking the voltage across the LEDs to make sure that the correct duty ratio is set for a specific potentiometer resistance.	Y
	3. It should have enough storage to contain our logic for controlling the LED drivers.	3. Make sure that the logic is able to be stored in the microcontroller and that the correct set of LED drivers are being turned on based on the five signals that are being sent from the microcontroller. This can be checked by making sure the infrared LEDs are correctly on when the switch for them is turned on and the white LEDs are on when the switch for white LEDs is turned on.	Y

	4. In the event of a fault, it should be able to turn off the LED driver PWM signals too low.	4. Have edge cases - test with overvoltage conditions to make sure that the LED drivers are being turned off in the event of a fault signal from the LED drivers. 5. Check the duty ratio produced by the microcontroller going to the LED driver interface. The 5V is around the nominal voltage expected across the potentiometer at its highest resistance setting. **$DutyRatio_{PWM}$ = $V_{potentiometer}$/5V**	N
LED Drivers	1. The LED driver most importantly should be able to achieve the desired switching frequency. We chose the 100kHz since it is high enough that the human eye cannot detect the flickering on and off the LEDs. This is achievable since the LED driver can support 100 kHz to 2.2 MHz. We can tolerate a relatively large tolerance on this since the human eye cannot reasonably notice a flicker when the frequency is over 200 Hertz.The frequency of the current output is at 400kHz with a higher tolerance of +/-10%.	1. We can verify this by measuring the output current waveform to LEDs using a current probe and an oscilloscope and then both manually and through the oscilloscope ensure that the current waveform has the required frequency of 400kHz.	White LED Drivers(N) Infrared LED Drivers (Y)

	2. Another important aspect of the LED driver is that it should be able to produce the duty ratio of the current waveform as requested by the microcontroller. When the LEDs are to be turned ON, the duty ratio can range from 1% to 100% of the period of the current waveform. The lowest that the PWM on time can go to is 150ns which is much lower than the 1% dimming at our switching frequency 100kHz so this will not be a constraint when designing our project. The duty cycle of the output current is the same duty ratio with a tolerance of +/- 1% as the input PWM signal from the microcontroller.	1. We can verify this by measuring the PWM signal generated by the microcontroller to the LED driver and then also measuring the PWM current signal produced by the LED driver on an oscilloscope and voltage and current probes. Then we manually compare the duty ratios on both these waveforms to see if they are similar to each other.	White LED (N) Infrared LED (Y)

	3. Finally the voltage and current supplied through the LED drivers should not exceed the maximum voltage and current ratings of the LEDs. a. For the infrared LED, the voltage supplied cannot be greater than 2V but lesser than 1.8V and the current supplied cannot be greater than 500mA. The LED driver has to ensure that these limits are not exceeded. b. For the white light LED, since each batch of 12 LEDs are arranged in parallel. Each white light LED can handle a maximum of 3V and a minimum of 2.9V and maximum of 120mA and a nominal current of 60mA.	Voltage supplied: Ensure that the voltage supplied to the LEDs are within the voltages the LEDs can handle. a. Infrared LED Driver: 1.8V to 2V b. White Light LED Driver: 2.9V to 3V Current Supplied: a. Infrared LED Driver: 350mA to 500mA b. White Light LED Driver: 60mA to 120mA 1. We can verify this by connecting voltage and current probes to the output signals to the LEDs and then connecting these probes to an oscilloscope where we can see if the voltage and current waveforms and within the tolerable values. 2. We will then do the above for different PWM width settings to ensure that the voltages and currents are still within limits. Some PWM width settings: 10%, 30%, 50%, 60 %, 70 %, 80%, 90%	White LED (N) Infrared LED (Y)

LEDs	1. The LED should be bright enough on the highest setting on the brightness knob to luminate the cancer cells. a. For the infrared LED: The radiant power should be between 100-200mW and this can be measured using a photometer. b. For the white light LED: The expected flux is 16-18 Lumens each. We can use a light meter to measure the Lumens.	For both these LED testings, we need to use a lightmeter to measure the lumens from the LEDs. This is especially important for the infrared LED since we are unable to see it with a human eye. The visible light LED can be manually also verified to see if the brightness is changing as we change the PWM width.	White LED (Y) Infrared LED (N)
	2. The wavelength of the LEDs should be within the desired limits and we can measure this using a spectrophotometer although this might be hard to obtain. a. For the infrared LED: The wavelength should be between 780nm and 785nm. b. For the white light LED: The wavelength should be between 400nm and 700nm.	We can verify the wavelength emitted by the LED by using a spectrophotometer although this will be a little difficult to obtain. The wavelength emitted by the white light LED can roughly be estimated to see if the color corresponds to a warm white light as rated or if it is a different color which would indicate a malfunction.	White LED (Y) Infrared LED (N)

Appendix C Intended Schedule

Date	Manogna	Jeremy	Yogavarshini
10/2	- Look into the logic of the microcontroller on how to send the SPI signal to the LED to get them to work as we want	-Create a rough draft of the PCB design that will get shipped out next to to give an idea on what should be on the PCB	-Look in the PCB Design where the white and infrared LEDs will be connected to determine what will be needed on the PCB
10/9	-Connect the user interface to the microcontroller on a dev board to test the SPI and start looking at edits for the round 2 of PCB design.	-Debug and Test the PCB design in time before the first round of orders. Finish at least one day before.	-Design and Test the PCB design for the LEDs. -Finish this in time for the first round of orders.
10/16	-Design Doc: Rewrite high level requirements -Design Doc: Rewrite UI and microcontroller and verifications in terms of a table -Make a list of parts for UI and microcontroller drivers -Recheck schematic if possible	-Design Doc: Redo power supply requirements -Design Doc: Rewrite battery and voltage regulators and verifications in terms of a table -Design Doc: Redo voltage regulators requirement -Make a list of parts for rechargeable battery and voltage regulators	-Make a list of parts for LED and LED drivers -Design Doc: Rewrite LED driver and LED requirements and verifications in terms of a table -Compile the list of parts and send out to ordering -Redraw images which are not clear in the design doc -Redraw block diagram with new voltage regulators
10/23	-Using the parts, test with a beadboard to make sure UI components are working like necessary -Debug and test the PCB design to see if there are any issues with pins and schematic -Individual Progress Report	-Use the ordered parts and test on a breadboard with an arduino uno -Make changes to PCB design if necessary to the PCB in time for the third round of orders -Individual Progress report	-Start with breadboarding of the LED and the LED drivers -Test and debug the LED drivers and the LED -Individual Progress report
10/30	-Solder parts onto the PCB	-Solder the parts onto the	-Solder and test the

	and make sure that the potentiometer and switches work together with the LEDs -If there any changes to the parts due to a change in design, make sure to order parts -Test the potentiometer values using probes to make sure values are consistent -Start putting all subsystems together	PCB when the PCB arrives -Test each pcb individually like the battery and the main pcb -If PCB arrives and not working, order one by ourself with plans of making it for the mock demo	LEDs and LED drivers on the PCB -Measure the duty ratios to the LED based on the UI input -Work on the code for the microcontroller
11/6	-Test the whole PCB and ensure that each part works with the other -If something fails, develop back-ups to supplement this	-Help with testing PCB as a whole -Prepare backups if things don't go as planned	-Work on testing all the systems together and make sure to debug any issues -Come up with a plan to fix if parts are not working well together
11/13	Reserved for last second changes that are arise during mock demo	Reserved for last second changes that are arise during mock demo	Reserved for last second changes that are arise during mock demo
11/20	Fall break	Fall break	Fall break
11/27	Prepare for the demo and presentation. Start working on the report.	Prepare for the demo and presentation. Start working on the report.	Prepare for the demo and presentation. Start working on the report.
12/4	Present presentation and finish the report.	Present presentation and finish the report.	Present presentation and finish the report.

Appendix D Inductor Formula

$$L_{min} = \frac{V_{out}(V_{in,max} - V_{out})}{V_{in,max} I_{LED} 0.2 f_{sw}}$$