Final Report

Group name: The Herb Growers

Group members: Salem Adda-Berkane, Cirino Costa, Owen O'Brien, James Sullivan

Section: A2

Professors: Zaman & Zosuls

Spring 2022 Semester

Executive Summary: The goal of this project was to design and build a device that maintains optimal lighting conditions for plants grown in windowsills. This product is primarily targeted towards those who cannot physically attend to their plants throughout the day. When coming up with our design, our group prioritized cost, structural integrity, and overall reliability. The product we ultimately created encompasses all three of these.

The final product utilizes a BH1735 light sensor, two potentiometers, a MOSFET, and a DS 3132 clock in order to automatically adjust the amount of light emitted while allowing for full user customization. Through the turning of one potentiometer, the user can adjust the amount of time they want their plant to be exposed to optimal lighting (choosing between 2, 4, 6, 8, or 10 hours). On the other hand, the turning of the second potentiometer allows the user to customize the intensity of said light (choosing between the light being off, emitting 2,000 lux per LED, 4,000, 8,000, or 10,000). In doing so, we were able to ensure that our product reaches or surpasses all plant light intensity and time requirements, with the ability for users to set limits on each. Furthermore, our design was also built with the reliability and durability of each of these aspects in mind. This means that our final product is made up of durable materials (i.e. wood, plastic, and metal) while also exceeding all time-based heat testing benchmarks by wide margin (which works to prevent components from burning out).

Introduction: The objective of the grow light was to provide optimal lighting for plants like thyme, mint, and basil. It needed to be an automatic light that could accommodate multiple different sized pots. It also needed to fit on a windowsill and provide flexible adjustments for the user. To provide optimal light, the device needed to supplement light for at least six hours with some plants needing more or less, such as mint which needed five.

Problem Statement: The goal is to design a device that automatically maintains optimal lighting conditions for indoor plants to reduce the difficulties of plant ownership for busy people.

Design Alternatives Considered: The group had many design options. For suspending the frame, the group considered the use of a tension bar or wall mounts. For attaching the light to the frame, the group considered the use of chains or bars. For frame material, the group considered wood or aluminum. For the circuit/brain, one of the decisions made was whether to use a resistor and potentiometer circuit or a MOSFET. Another decision relating to the brain, was whether a hardware clock or software arduino delays should be used to regulate time. Regarding the interface and user input, the use of an LCD screen and two potentiometers were considered. For the lighting, individual LEDs and premade light strips were considered. To focus the light, LED lenses were considered. For powering the lights, a battery and a wall outlet were considered.

Basis for Design Selection: After careful consideration of all the alternatives, the design was finalized. Of particular note were the design decisions regarding the light, and power sources. Regarding the light source, an array of individual LEDs was selected in favor of LED strips. The array was selected, as it provided greater flexibility than the light strips in the amount of light outputted (due to the ability to add or subtract individual lights) and the light coverage (due to greater freedom in light location placements). In total 21 LEDs were used, with each being blue, and requiring 3W and 0.7A. Blue was chosen because it stimulates optimal plant growth. Initially, these LEDs were going to be focused via 60° lens, however testing proved that the impact of the lens were negligible and so they were removed. The entire array was powered via a wall outlet. Specifically, a power supply was used to draw electricity from an outlet. The supply was rated to regulate an output of 12W and 4A. This power supply was chosen as the power source because it was determined that the LEDs would need 12W if placed in 7 parallel branches (with each branch consisting of 3 LEDs).

In regards to the support system, wall mounts were chosen over a tension rod. This choice was due to a tension rod likely not being able to hold the weight of the finished device. When considering a chain vs bar support for holding up the lights, adjustability was prioritized because plants vary in height and growth. This prioritization took the form of the chains which allowed easy movement/adjustment when compared to the bars. For the frame, the group considered wood and aluminum. Wood was selected due to its cheap price, ease in use, and lightweight. In contrast, an aluminum frame would have been too heavy to suspend.

Description of Final Product: The final product consists of three distinct parts, the frame, the light, and the brain. The frame consists of two wall mounts, a beam, and 8 hooks (4 on each side of the beam). The brain is mounted on top of the beam, while the light hangs from the hooks via 4 chains. The beam and wall mounts are seperate parts to allow for the mounts to be adjusted based on the window to which the device is being mounted. The inclusion of 8 hooks serves a similar purpose for the height and placement of the light.

The brain consists of an Arduino Uno, a BH1750 light sensor, a DS3231 clock, two potentiometers, and a MOSFET. The Arduino Uno acts as the controller for the light with various adjustments being made to it, based on the sensor's readings and user input. As expected, the light sensor detects the light levels in a room. Working in concert with the clock, it triggers the light to turn on, if the room's light is too low and the time is between 0800 and 1600. The two potentiometers act as the source of user input. One of them controls the light's intensity whilst the other controls the time that optimal light should shine on the plant. Both of them are stepped, with 5 options. The time potentiometer has the options of 2, 4, 6, 8, or 10 hours. The intensity potentiometer has the option for the LEDs to emit no light, 2000 lux, 4000 lux, 8000 lux, or 10000 lux. Once the Arduino deems the proper conditions have been met (the time is between 0800 and 1600 with the light levels having been above 100 lux for a time less than that inputted), then a signal will be sent to the MOSFET to allow the circuit to be closed.

The light is composed of 21 LEDs which are divided into 7 parallel branches. Each of these branches consists of 3 LEDs and a 3Ω, 5W resistor. Each LED sits on a heat sink, with the entire array being attached by screws to rails on a larger aluminum holder. The top of the holder features four mounting screws which are used to hang the chains that hold up the light.

Evaluation of Results: The metrics established at the start of the project were all fulfilled to some capacity, some went beyond the stated metrics while others met the minimum requirements. The first metric involved providing optimal light for maximum plant growth. The 21 LEDs summed to a total theoretical lux rating of 45,208.422 lux when covering an area of 0.25 squared feet. Due to the product being a supplement to the natural sunlight, the levels of lux that each LED emits levels between 2000-10000 lux. The maximum average lux level that can be reached is approximately 40,000 lux, depending on variables the user can change such as distance to plant and LED intensity. Based on power levels measured over time, this maximum will not result in the product burning out. The structure fulfills the size metric by fitting in an area described as 2.667 squared feet. Finally, the chains on the lamp allow for manual adjusting of lamp distance to plant in order to allow for unrestricted plant growth and to accommodate all kinds of houseplants.

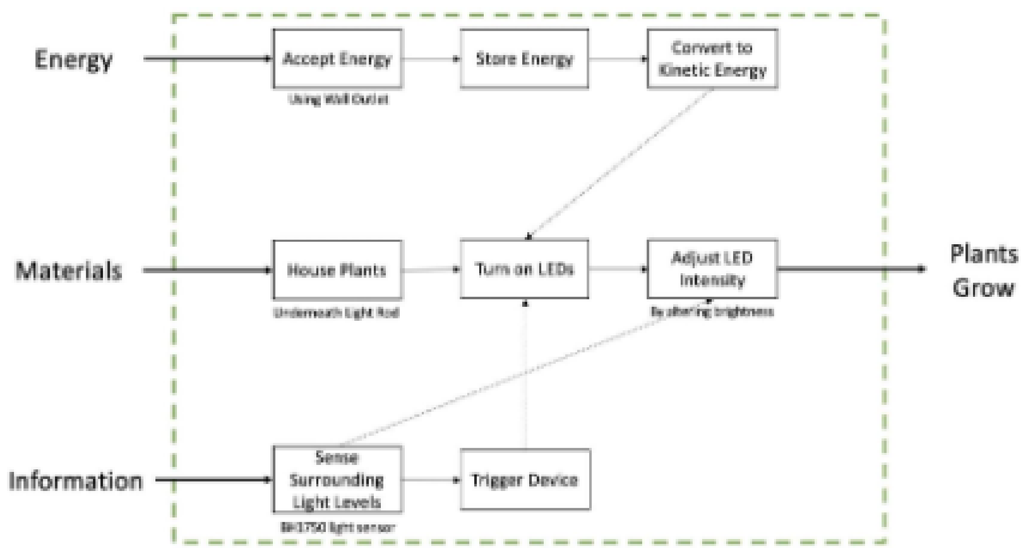
Appendix:

Indoor Grow Light

Means / Functions	1	2	3	4
Light	MH	HPS	LED	Fluorescent
Light sensor	Photo-Diode	Photo-Transistor	Photo-Resistor	Photovoltaic Cells
Attachment	In soil	Wrap around base	On table/stand	Under Pot
Adjustable Height/Stand	Unistrut	Clamp	Screw	…
Adjustable Light Intensity	Variable Resistor	Change Voltage	…	…

Notes:
MH: Metal Halide bulb
HPS: High-Pressure Sodium bulbs

Means and Function Table.



Glass Box Diagram.

Pairwise Comparison Chart

	Provides Optimal Light	Automatic Light Adjustment	Accommodates at least a 6 in pot	Accommodates 3 plants at once	Fits on windowsill	Provides flexible adjustment for user	
Provides Optimal Light	X	1	1	1	1	1	5
Automatic Light Adjustment	0	X	0	1	1	1	3
Accommodates at least a 6in pot	0	1	X	1	1	1	4
Accommodates 3 plants at once	0	0	0	X	0	0	0
Fits on windowsill	0	0	0	1	X	1	2
Provides flexible adjustment for user	0	0	0	1	0	X	1

Pairwise Comparison Chart.

Parts	**Number of Parts**	**Price ($)**
Wall Mount	2	N/A
2x4	1	7.26
Component Box	1	N/A
Box Lid	1	N/A
Yellow Plastic Chain	1	30.00
Steel Hooks	8	10.00
Nuts and Bolts	60	3.00
Washers	16	2.00
Arduino Uno	1	24.20
DS3231 Clock	5	18.49
Potentiometer	2	8.49
3W LED	21	11.85

Star Style Heat Sink	21	4.50
MOSFET	1	9.99
Power Supply	1	13.80
BH1750 Light Sensor	1	5.99
Total:		$149.57

Final Part List.

Code Flow Chart.

<u>definitions/imports{</u>

1. Import/Prepare Time library for use by code. Time library is used to track the time of day in order to ensure that the light does not turn on outside of growth period (i.e the middle of the night).

#include <Time.h>

2. Import/Prepare BH1750 and Wire libraries. BH1750 library is being imported as it is necessary to use the light sensor. The Wire library must be imported as the BH1750 library is contingent on it.

#include <Wire.h>
#include <BH1750.h>

3. Set various pins for the sensor, potentiometer (exists for user making adjustments to LED intensity), and voltage regulator (device that will turn lights on and off, needed because LEDs will draw power from the wall with the arduino acting merely as a gatekeeper which determines whether or not electricity will be allowed to flow).

BH1750 LightSensor;
int Potentio = A0;
int Voltage_Regulator = 2;

4. Create variable Cycles which will be used in if-statement further down in the code to determine whether or not a light level drop is a temporary fluke (i.e something causing temporary light level drop, like a cloud passing by, or a more long term drop, like an overcast day). Also creates a variable LightLev which stores the value of the potentiometer which is used to control the intensity of the lights.

int Cycles=0;
int LightLev=0;

5. Create a variable that will be used to check when the light turns on. The intention is to use it to track the amount of "optimal light" that the plant has received.

float start_time=0.0;

};

setup(){

1. Begins serial reading, so that the sensor's light level is read. This is done for the sake of storing a value which will later be checked against in if statements (this is used to determine if the light is within acceptable parameters or not).

Serial.begin(9600);

2. Initialize Wire functions because BH1750 is reliant on them (but does not initialize it automatically), and then the light sensor.

Wire.begin();
LightSensor.begin();

3. Configure pins for the Voltage Regulator and the potentiometer. The Voltage Regulator will be outputted to, whilst the potentiometer, must be read and so will be taken as an input.

pinMode(Voltage_Regulator, OUTPUT);
pinMode(Potentio, INPUT);

4. Set the current time that will be used further on in the code in order to check whether or not the light is on within the appropriate growth window. It should be noted that it is likely that an arduino-compatible clock will have to be added to the system.

time_t current_time= now();

}

loop(){

1. Read the value of the light sensor. Note it does so as a float to allow for greater gradiation in reported reading (as the light sensor can precisely read up to 0.5 lux).

float lux = LightSensor.readLightLevel();

2. Read the value of the potentiometer as set by the user. This value is gotten by reading the angle of the potentiometer. This angle is then converted to a voltage equivalent, before being saved as the variable Brightness. This variable is used later on when setting the intensity of the light.

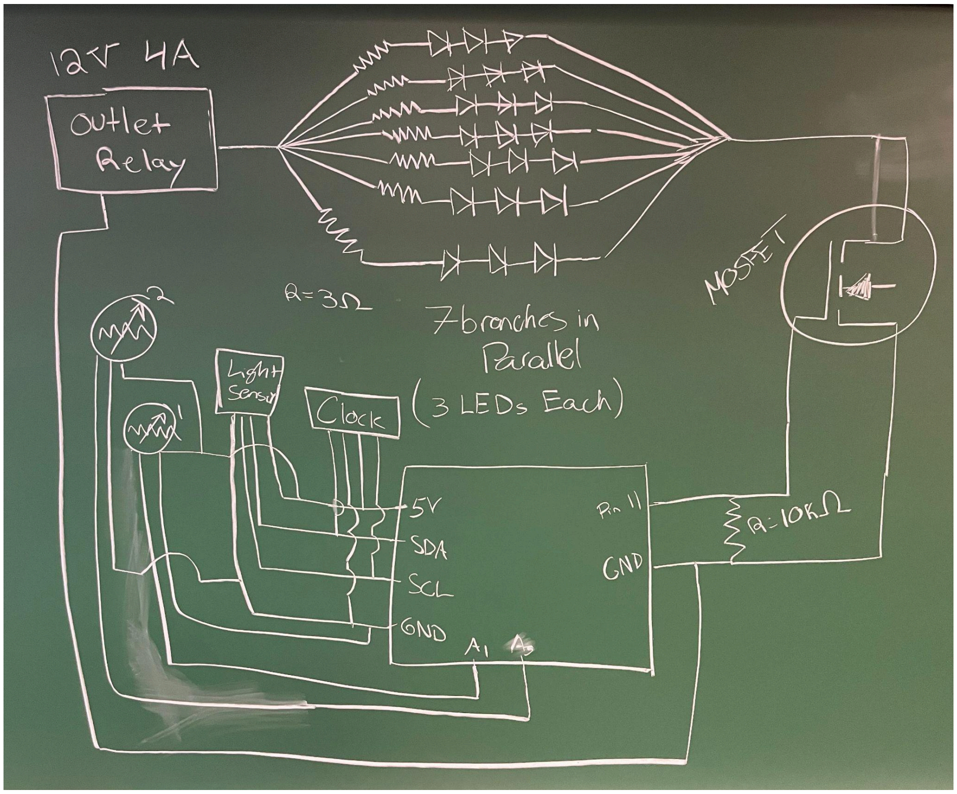
```
LightLev= analogRead(Potentio);
Brightness= map(LightLev, 0, 1023, 0, 255);
```

2. If statements to check whether or not the light level is adequate. If the light level is initially determined to be too low, then it is checked twice more before turning on the light. The gap between checks is about 2 minutes. This is done to prevent "false starts" to the LEDs. The two checks are tracked by adding to the variable Cycles. When Cycles==3 (as in three checks of the light level have been made and each check has determined the light to be too low), the light will turn on. If the light level is determined to be above the set threshold, then no change is made. It should be noted that when the light does turn on, it records the time. This is done to monitor the amount of "optimal light" that the plant receives. Once the light is on there is a delay of 30 minutes before the light levels are checked again.

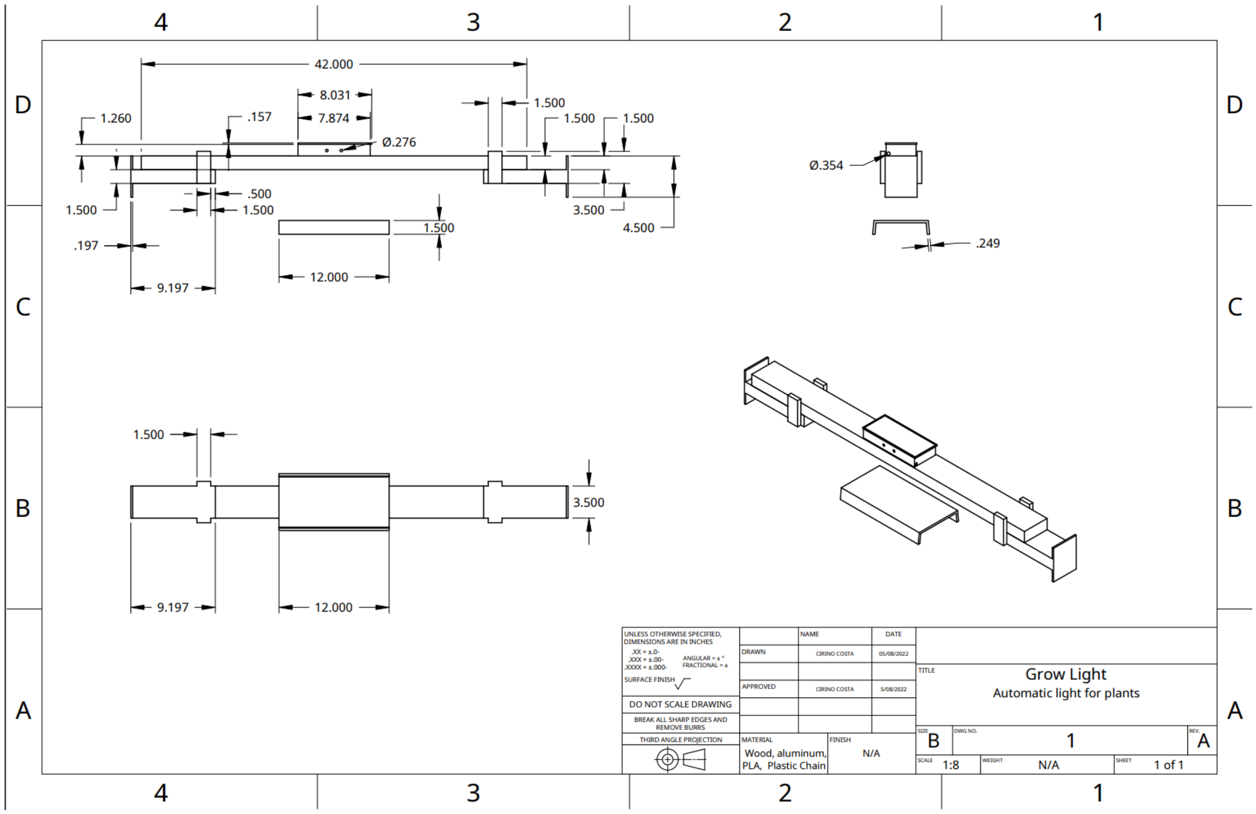
```
if (lux<=Light_target) && (Cycles==3) && (start_time<=8){
analogWrite(Voltage_Regulator, Brightness);
start_time=hours(current_time) + (minutes(current_time)/100);
delay(1800000);
}
else-if (lux<=Light_target) && (Cycles<3) && (start_time<=8){
Cycles+=1;
delay(120000); //roughly 2 minutes between checks
}
else{
analogWrite(Voltage_Regulator, Brightness);
start_time=hours(current_time) + (minutes(current_time)/100);
delay(1800000);
}
```

3. Turns off light and stops checking light levels if outside of acceptable growth time (assumed to between 18:00-06:00).

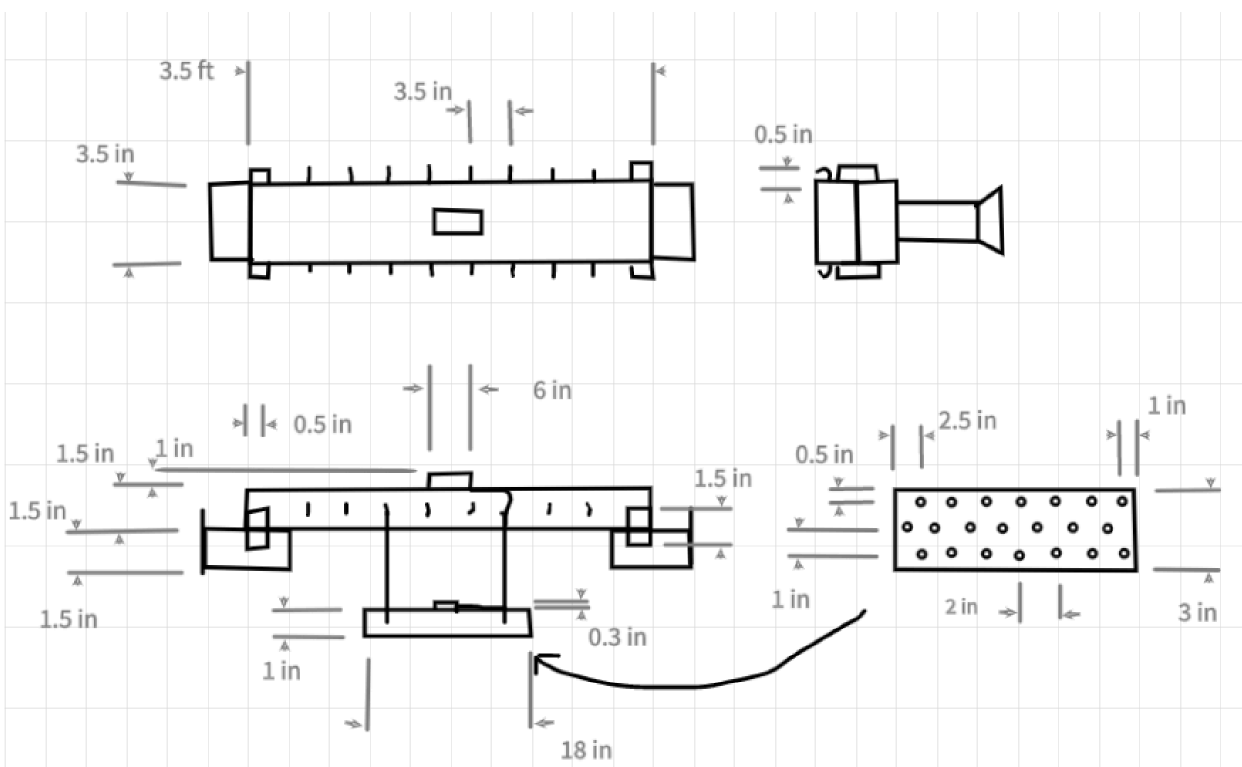
```
if (hour(current_time)<=18) || (hour(current_time)>=6) {
analogWrite(Voltage_Regulator, LOW);
delay(43200000);
}
}
```



Electric Circuit Diagram.



CAD for Frame.



Initial Sketch of Frame and Light.

Power Budget

Name	Voltage (V)	Current (A)	P=VI (W)	Duty Cycle	Total Power (W)	Contingency	Grand Total Power (W)
Power Converter	12	4	48	100%	48	0%	48
Arduino	5	0.1	0.5	100%	0.5	30%	0.65
Sensor	4.5	0.007	0.0315	3.3%	0.00104	20%	0.00125
LED (Individual)	3.57	0.600	2.146	100%	2.146	25%	2.6825
LED (Branch)	12	0.600	7.2	100%	7.2	25%	9.0
Arduino Clock	3.3	0.040	0.132	100%	0.132	25%	0.165
MOSFET	12	4	48	3.3%	1.584	20%	1.9008

Power Budget Table.

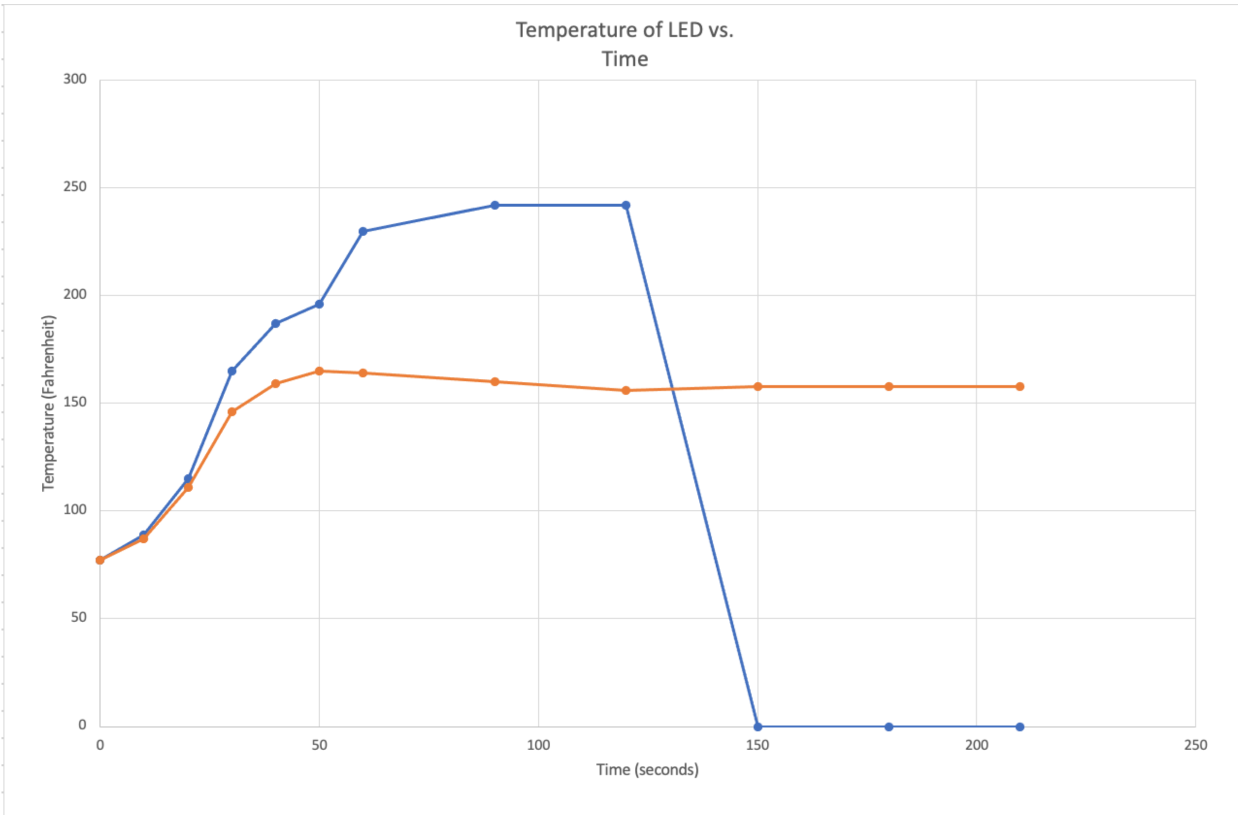
Power	**Intensity**	**Distance(in cm)**
0.9	0.012704	10
0.76	0.010727	11
0.66	0.009316	12
0.56	0.007904	13
0.45	0.006352	14
0.38	0.005364	15
0.34	0.004799	16
0.3	0.004235	17

0.27	0.003811	18
0.24	0.003388	19
0.21	0.002964	20

Intensity vs Distance Table.

Time (seconds)	Temperature Without Heatsink (Fahrenheit)	Temperature With Heatsink (Fahrenheit)
0	77	77
10	89	87
20	115	111
30	165	146
40	187	159
50	196	165
60	230	164
90	242	160
120	242	156
150	0	158
180	0	158
210	0	158

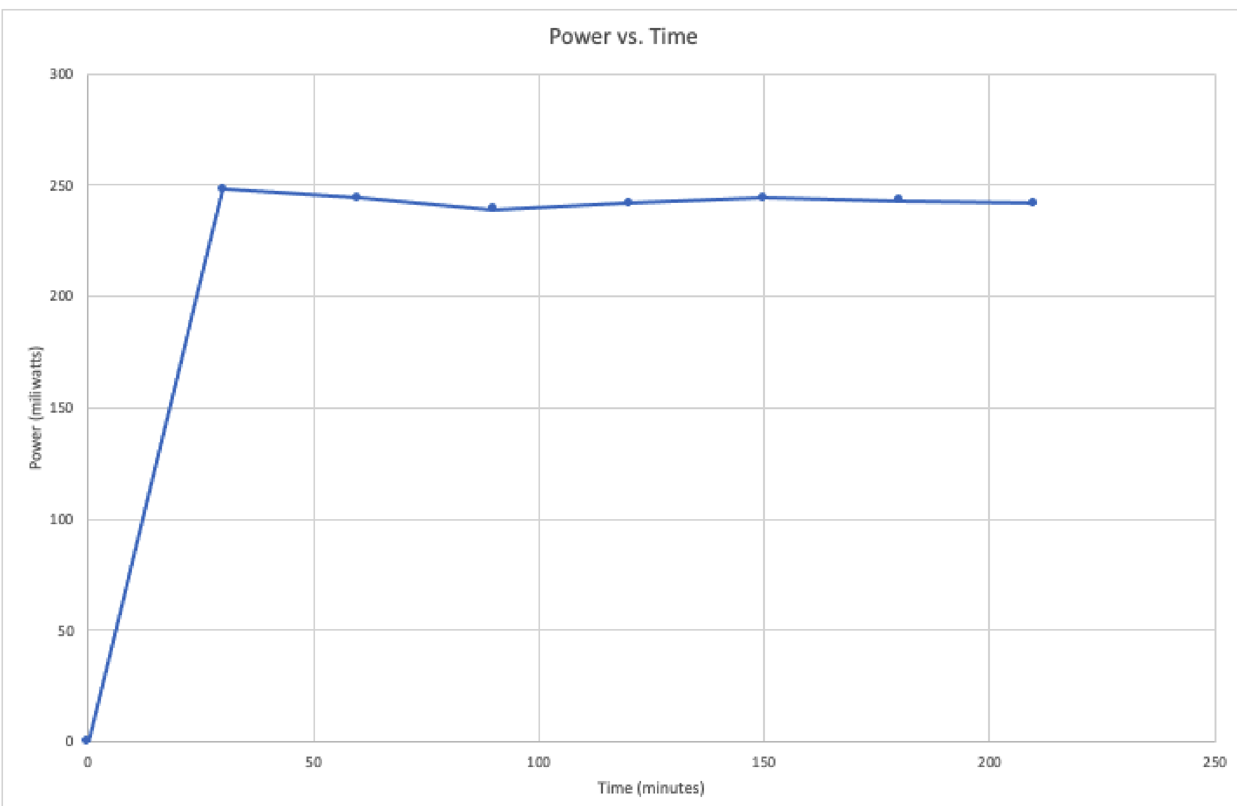
Table for Temperature of LED with and without Heatsink



Graph for table above. *Longest run time recorded for LED with heatsink was 1 hour and 25 minutes with no temperature change from constant 158 degrees fahrenheit. Orange is with a heatsink and blue is without.

Approximate Power Level (miliwatts)	Time (minutes)		Distance (0.5 Feet)
0	0		
248	30		
244	60		
239	90		
242	120		
244	150		
243	180		
242	210		

Power vs Time Table.



Power vs Time Graph.