Lava Lamp 2.0 :
The Inductioning

By:
Erik Eldridge
Ignacio Iber
Jake Lawrence

Final Report for ECE 445, Senior Design, Spring 2017

TA: Daniel Frei

2 May 2017

Project No. 44

Abstract

The lava lamp is an iconic piece of popular culture which was originally developed in the 1960's. Over time it has become a novelty decoration for a room in an average home. The lamp's design has remained largely unchanged since its inception. The Lava Lamp 2.0 project is a recreation of the traditional lava lamp that brings four major enhancements, making it faster, brighter, interactive and safer for the user. In this paper, we will demonstrate how the lava lamp accomplishes this using induction heating with a decoupled lighting system.

Contents

1. Introduction

Lava Lamp 2.0 is the second iteration of a series of technological improvements made to the classic lava lamp. Preceded by Lava Lamp 1.0, our project makes further improvements to the flow time, brightness, interactivity, and safety of the lava lamp.

1.1 How a Lava Lamp Works

The lava lamp globe contains two liquids: one is mainly water and the other wax. On the bottom of the lava globe there is a heating element that heats up both liquids just at the lower part of the lamp; whereas the upper part stays cooler, since there is no heating source.

As both liquids get hot, only wax changes its density: warming up, the density drops (under the water's density) and therefore the wax goes up; cooling down, the density rises (over the water's density) and therefore the wax goes down again.[1]

This density difference principle (explained in Figure 1) is what makes the lava lamp flow and form bubbles that rise up and down continuously as the wax warms up and cools down.

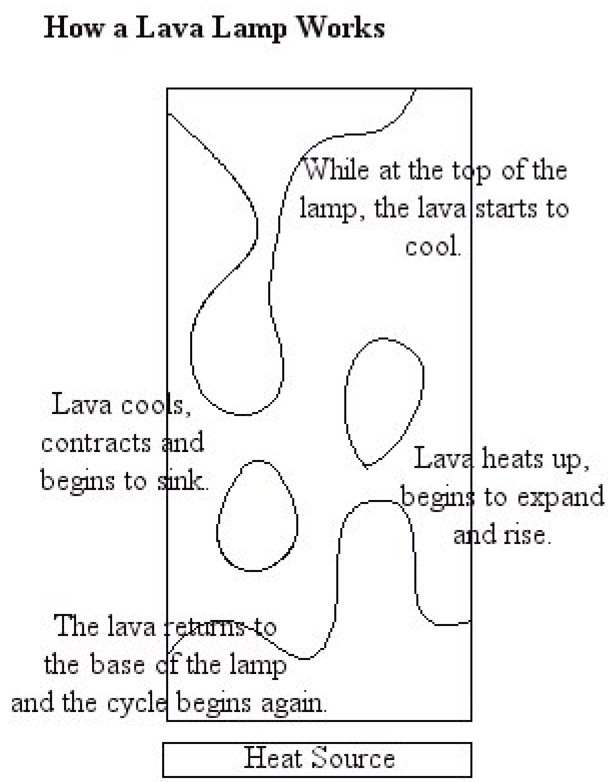

Figure 1: Explanation of lava lamp fluids.[2]

1.2 Background

The classic lava lamp has the most basic design, consisting only of a 25 W bulb positioned underneath a glass lava globe. This bulb serves the purposes of both heating and lighting the lamp, albeit inefficiently.

Because it gives off relatively little ambient light, it acts as a poor light source for illuminating a room. Relying on only the latent heat from the bulb, the time of lava flow from the start of operation is upwards of two hours. This is an inconvenience for those who would like to use the lamp on a whim. Furthermore, the classic lamp has only an on and off setting with no variations of brightness or color. The lamp gets hotter the longer it is on, presenting a safety hazard for those touching the lamp or removing the globe from the base.

Lava Lamp 1.0 made improvements to the classic model, decoupling the lighting and heating mechanisms to allow temperature control without affecting the amount of light generated. The temperature feedback control allows the lamp to stay at 40∘C and not get hotter. It utilizes a 100 W heating resistor array to decrease the flow time to 25 minutes, down from 2 hours. It added user interactivity with five different color settings and variable brightness. Illuminated by two RGBW (red, green, blue, white) LED's (light-emitting diodes) instead of a 25 W bulb, it aimed for increased brightness compared to the classic lamp but unfortunately fell short.

1.3 Motivation and Purpose

Our project addresses the problems of the lava lamp 1.0 by first improving the LED light source. Our light source has been upgraded to an array of three RGB color LEDs and three white LEDs. this new lighting system allows for a substantial increase in brightness, which allows the lava lamp to function as an actual light source for a room. this innovation also allows user control for brightness and color. The problem of slow heat up time will be solved by our new induction heating system. This replaces the heating plate as a heat source. The induction system directly induces current into a metallic workpiece, located inside the lava globe, which will heat the lamp to operational temperature much faster. It will also improve user safety because it will keep the glass cooler and remove any external hot surfaces that could potentially cause a burn.

1.4 Objectives

To upgrade the previous versions of lava lamps, this project will focus in the following main four features to improve:

- Operation: Reach lava flow at operational temperature (50°C)[3] within ten minutes.
- Brightness: Obtain 3000 lux (unit of illuminance) one foot away from the light source.
- Interactivity: Switch between seven different colors and five modes of brightness.
- Safety: External surface of lava globe does not exceed 45 °C [4].

2 Design

2.1 Block Diagram and Descriptions

Our lava lamp consists of five well differentiated parts and the input, shown in Figure 2. The power block supplies DC (direct current) power to the energy block and to the control block. The PCB (printed circuit board) is in charge of controlling the amount of power delivered, as it receives feedback from our measurement block. This measurement block includes a temperature sensor to prevent overheating. The energy block is made up of the induction coil (electric to electromagnetic energy), the workpiece (electromagnetic to heat) and the multicolor LEDs (electric energy to heat). We can fulfill our four high level requirements because the heat flow will only occur inside the glass, the LEDs can be as bright as a traditional lava bulb, the MCU (micro-controller unit) controls the different colors and brightnesses, and the induction will lead to faster heating.

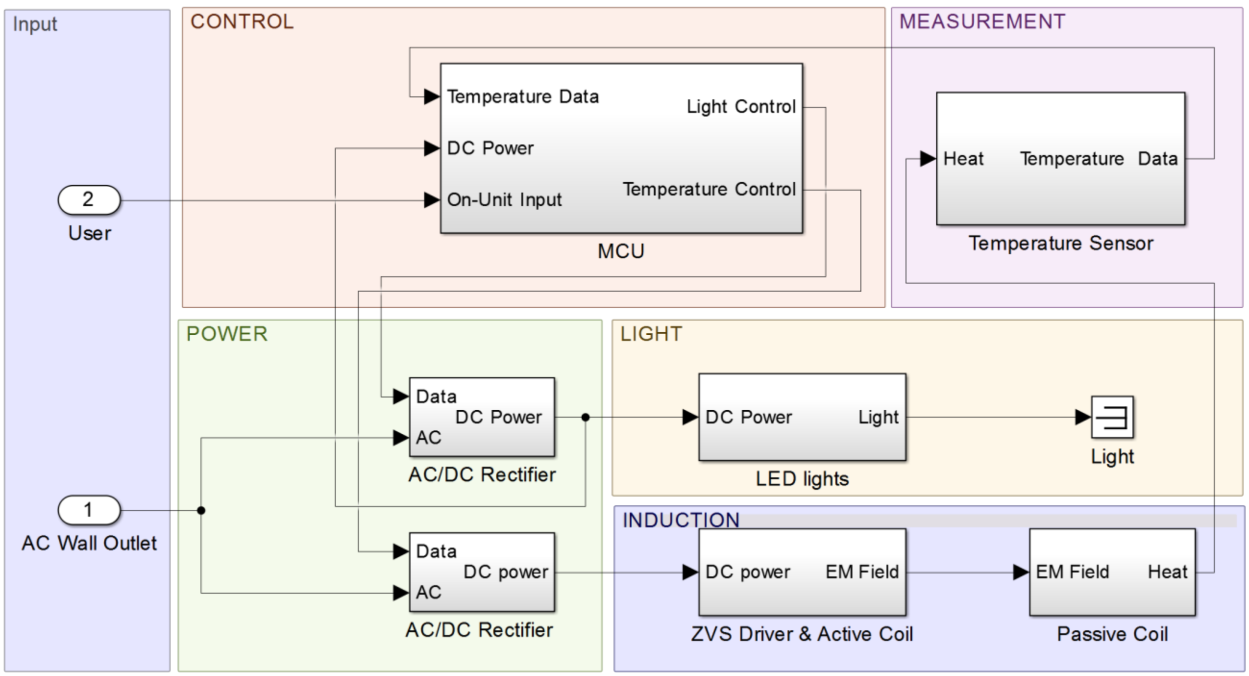

Figure 2: Block diagram of the lava lamp system

2.1.1 Input

The input block remains largely unchanged from our design review. To power the entire system, 115 V, 60 Hz (hertz) AC (alternating current) is sourced from a wall outlet and fed into the power block. The power block then converts this input voltage into desired values to power subcomponents of the system.

For user input, there will be a set of pressure-sensitive buttons and an ON/OFF switch mounted to the base metal part of the lamp. There will be a button to cycle through seven LED colors (instead of three as previously stated in the design document) and a button to cycle through five LED brightness settings.

2.1.2 Power

2.1.2.1 PCB Power Supply

The power block is separated into two independent AC to DC converters. One is to power the PCBs and the other is to power the induction system. The PCB power supply consists of a 115:6.3 turn transformer that converts voltage from a wall outlet to 6.3 V AC . A bridge rectifier and filter capacitor is then used to convert the 6.3 V AC to a smoothed 7.3 V DC signal. This smoothed DC signal will be the primary power supply for both PCBs. The tolerance for this value is fairly tight because the resistor values on the input of the LEDs were specifically chosen to work with this voltage. Our original design called for a 12:1 turn transformer that would convert the wall voltage to 10 V AC, but design choice for the LED system caused us to change our transformer.

2.1.2.2 Induction Power Supply

The induction power supply is the other subsection of the power block. Like the PCB power supply, it utilizes an AC to DC converter. the wall voltage is converted to 24 V AC by means of a 115:24 turn transformer. This power converter, like in the PCB power system, utilizes a bridge rectifier and filter capacitor to convert the 24 V AC into 31.5 V DC. This DC directly powers the induction system. In our design review, we planned on using the rectified DC from the PCB power supply to also power the induction system, but through research it was determined that much higher voltage was required. The design choice was then made to have a completely independant power supply, as described, to operate the induction system.

2.1.3 Control

2.1.3.1 Microcontroller Unit

The control block receives input from both the user and the temperature sensors, the MCU translates that input to a different output, changing the value of the MOSFETs (metal oxide semiconductor field-effect transistor) gates. The signal from the temperature sensors is between 0.1 and 1.75 V, depending on that input the heat MOSFET gets high (3.3V) or low (0V). The PWM signals for each of the LEDs color adjust the brightness of the four colors (red, green, blue, white).

2.1.4 Light

2.1.4.1 LED Circuit

The LED Circuit receives four PWM (pulse-width modulation) signals from the MCU and receives up to 12.7 W from the PCB power supply. The PWM inputs are connected to the gate pins of four PSMN022-30PL n-channel MOSFETS. Three CREE XM-L RGBW LED's are used to output color (white diode is ignored), and three CREE Single-Die XM-L White LED's are used to output ultra-bright white light at 6000K color temperature. Each MOSFET controls the operation of a single color channel (either red, green, blue, or white) and is connected at the drain to every CREE LED which outputs that color. Connected in series with each of the RGB and Single-Die white LED's are current protection resistors which are rated 13 Ω/ 2 W and 6.8 Ω/ 4 W, respectively.

2.1.5 Induction

2.1.5.1 ZVS Driver and active coil

This block is operated by an oscillator circuit connected to the active coil, which generates an alternating magnetic field which will induce eddy currents to produce heat in the workpiece. The power supplying the coil will be controlled by the MCU, which will react based on info received from the temperature sensor. The MCU will provide a signal to the gate of a MOSFET to act as a switch. Enabling the MOSFET switch will provide full power to the induction driver, while disabling the switch will turn it off. The original design in the design review called for the MCU altering the feedback loop of a buck converter to control the supply voltage to the induction heater. Because a high power buck converter proved to be too expensive for us to implement, the on-off design was chosen as an alternative.

2.1.5.2 Workpiece (passive coil)

The workpiece is the metal piece attached to the bottom of the lamp that is heated by a flat induction coil. This, in turn, heats the liquid at the bottom of the lamp and causes the lava to flow. This workpiece should heat the water in the lamp to operating temperature, 50 °C, within ten minutes. This design remains unchanged since the design review.

2.1.6 Measurement

2.1.6.1 Temperature Sensors

The TMP36 temperature sensor is mounted to the outside of the glass just below the induction chamber. This is the hottest part of the lava lamp besides the conductor itself, however this area is not accessible to touch. The sensor takes a continuous analog temperature reading from this area and provides continuous feedback to the control system via a voltage signal to the MCU.

2.2 Design Procedures

2.2.1 PCB and Induction Power System

We use a standard 120 V AC, 60 Hz wall outlet as our power source to supply power to both the lighting and induction systems. A separate power circuit is used for both of these systems because at the beginning of the project the current draw of our active coil and workpiece had yet to be experimentally determined. However, both circuits convert power in the same way. The first component of these circuits is a transformer, which steps the AC input down to a smaller, more usable value. Second, we use a bridge rectifier to provide full wave rectification which converts this stepped-down AC to DC. Then, we smooth the DC voltage with a capacitor that is placed in parallel with the load. This three-step system, shown in Figure 10 in appendix E, produces a stable DC voltage level with minimal ripple voltage. In our design, we used two separate AC to DC power converters to power the induction system and the LED and MCU system. An alternative approach would to use a single power converter and buck converters to supply necessary voltage levels to different components. However, high power buck converters are very expensive. While this alternative design is simpler and more compact, it was too expensive for us to implement.

A 3.3 V linear regulator was used to further step down the DC voltage output voltage of the LED power supply in order to power the microcontroller. A linear regulator is a relatively inefficient way to convert power, but because the microcontroller requires little power to operate, a linear regulator is perfect for supplying it. Linear regulators also provide a very smooth DC output, which is crucial for a microcontroller because a smooth DC input provides an excellent reference voltage for the microcontroller's various operations.

2.2.2 Microcontroller Unit

To control the whole lava lamp we decided to implement a Microcontroller that can interact with the user and the lava lamp temperature. The MCU chosen was the ATMEGA328P since we were more familiar with the programming procedures. To interact with the lamp our decision was to use two push buttons which control color and brightness, separately. Another option would have been a potentiometer for the brightness, but we considered that a push button would give more precision to the quantity of light, as it directly toggles to the brightness amount and the difference is more appreciable. The heating control helps the safety purpose we expected. Nevertheless, we could make it better by also controlling the induction heating with an external device to the MCU, to make it safe in case the MCU stops working.

In Figure 9 (Appendix E) we can appreciate the circuit schematic for the MCU, that requires a 3.3 V supply to work. Connected to it are the two buttons, two temperature sensors, the external clock (crystal oscillator and two 22 pF (picofarad) capacitors), the reset button, the heat circuit MOSFET as well as outputs for the 4 PWM signals of the LEDs (Red, Blue, Green, White) and Arduino outputs for programming. A flowchart is shown in Figure 20 Appendix G, to understand the functionality of each of the parts connected to the MCU.

2.2.3 LED Circuit

Brightness and interactivity are two main objectives of this project so our LED circuit is an important one. We were satisfied with the interactivity displayed with the CREE XM-L RGBW LED's used in Lava Lamp 1.0, but we needed more brightness so we decided to use three of them. Additionally, we chose to use three dedicated ultra-bright single-die white LED's in order to meet our objective for increased brightness. Each single-die white LED supplies 300 lumens at a typical operating current of 700mA for 900 total lumens of light. This is comparable to 865 lumens at 1 ft (see Section 2.3.3) in theory. However, after experimentation our three white LEDs yielded 3655 lux at 1 ft, satisfying the requirement (see Section 3.3.1). An alternative approach would be to use LED strip lights, however these contain lots of pre-built circuitry so we would be unable to fulfill our interactivity objective. The CREE LED's are simply a diode mounted on a small PCB, therefore we are able to adjust the brightness and color independently through separate MOSFETs acting as a switch for PWM control.

Additionally, we chose to use resistors in series with each diode for current protection purposes. The resistances are chosen such that the current through each color diode is near its 'typical' value of 350 mA for RGB [5] and 700 mA for white [6]. We chose to use equal resistors for red, green, and blue despite red drawing higher current due to the fact that red appeared dimmer. A higher resistance was chosen for white because of availability, though a lower value would have been better for brightness. The power draw for each LED is shown in Tables 4 and 5 Appendix D and the total power draw is shown in Table 6 Appendix D. Figure 18 Appendix F shows that the red color is among the dimmest, despite increased current draw. Power dissipation values well above the power dissipation requirements calculated in Section 2.3.3 were chosen to allow for cooler long-term operation.

2.2.4 Induction

The induction system is the main heating source for the lamp. this system consists of a DC powered inverter driver connected to a pancake shaped active coil. This pancake coil then creates an alternating magnetic field inside the work coil that repeatedly magnetizes and demagnetizes the workpiece. The combination of active coil and workpiece acts like a transformer, where the active coil is the primary and the workpiece is a shorted secondary. This arrangement causes large currents, called eddy currents, to flow through the workpiece and heat it up. A property called the skin effect causes most of the current to flow on the outer edges of the workpiece. In our application, the hot workpiece is used to heat the surrounding water of the lamp and cause the wax to flow.

2.2.4.1 ZVS Inverter

A ZVS (zero-voltage switching) inverter was chosen to power the active coil and create the alternating magnetic field. A ZVS inverter, as shown in figure 10 in appendix E, is a relatively simple circuit that can efficiently oscillate large amounts of power. When the circuit is powered on, voltage reaches the gate of each MOSFET (metal–oxide–semiconductor field-effect transistor) simultaneously. At the same time, current tries to flow through the drain of each MOSFET. The internal differences between each MOSFET causes one to turn on first. This, in turn causes all of the current to flow through the MOSFET in the on-state, while the other MOSFET simultaneously turns off. The MOSFETS then oscillate on and off to create the alternating magnetic field in the active coil. The oscillation is created by a parallel LC (inductor-capacitor) resonant tank. The pulldown resistors and zener diodes allow the MOSFETs to switch with zero volts across them, meaning they dissipate as little power as possible. The inverter is controlled by a separate MOSFET, which has its gate voltage supplied by the MCU. This acts as an on-off switch that controls supply of the input voltage to the inverter, which is how the lava lamp temperature is regulated.

An alternative design that was considered was using a royer oscillator instead of a ZVS driver. The royer oscillator is a simpler circuit to construct, but dissipates more power in its MOSFETs because it ignores the extra diode and pulldown resistor that make the MOSFETs switch at zero volts. This makes the royer oscillator less robust and less efficient than its ZVS driver counterpart.

2.2.4.2 Active coil

The active coil for the induction heating system is a concentrically wound pancake coil made from ¼ inch copper tubing. The coil we constructed is shown in figure 7 in appendix B. Copper tubing was chosen because of its sturdiness, relatively low resistance, and resistance to heat. Ideally, the active coil is to remain cool to the touch, however we were able to accomplish this. Other designs of active coils made from copper pipe utilize water cooling at high voltage, but we lacked the equipment to implement this.

Alternative designed considered involved changing coil material and changing coil design. The ideal design change would be using a high gauge litz wire, which is a very low resistance type of wire made with individually insulated strands of twisted copper. This would theoretically accomplish the goal of keeping the active coil cool to the touch. Another design alternative relies on using a center tapped pancake coil. This design eliminates one inductor in the ZVS driver, but can create excess stress on one of the MOSFETs if the center tap is not exactly centered. The final alternative is to use a formed round coil, which is in the shape of a cylinder, instead of a pancake wound coil. This design is more effective because all of the magnetic field is concentrated in the center of the coil. Using this design would require an alternate glass design for the lamp, which was out of the scope of this class.

2.2.4.3 Workpiece (passive coil)

The workpiece is the metal object, located in the base of the lamp, that becomes warmed by the induced magnetic field of the active coil. We chose stainless steel washers as the workpiece because they have holes to allow light to pass through from the LED's and because they fit through the opening at the top of the lava lamp. An alternative design would be to use a steel plate with holes drilled where the LEDs would be located. This would allow light from the LEDs to pass through to the lamp. For this to be feasible, we would need to be able to be able to alter the glass design to fit a plate in the top. As mentioned in 2.2.4.2, this is outside the scope of this class. Other materials besides stainless steel such as copper or aluminum could be used for a workpiece, which would alter the required resonant frequency and ability to heat up. Stainless steel has good heating properties, is easily obtainable, and is cheap. Because of these factors, stainless steel was the prefered choice.

2.2.5 Temperature Sensors

The measurement block of the lava lamp obtains information from two temperature sensors: one placed at the bottom of the globe which should reach the desired operational temperature (50 °C), the second one placed next to the heating circuit, to ensure its temperature doesn't exceed 45 °C. The temperature sensors are fed with a 3.3 V voltage and have an output voltage signal linear with the temperature.

2.3 Design Details

2.3.1 Power System Design Details and Calculations

The circuit design for our AC to DC converter is shown in figure 10 in appendix E. As described above in section 2.2.1, this rectifier circuit converts AC voltage to DC by means of a full wave rectifier IC (integrated circuit) and a filter capacitor. To find an appropriate capacitance for the power supply filter capacitor we use equation (1). In this equation, i represents max current through the system in A, t represents the half cycle time for the input AC voltage in seconds, ΔV is allowed output voltage ripple in volts, and C is resulting capacitance value in farads[7].

$$C = \frac{i * t}{\Delta V} \tag{1}$$

2.3.1.1 MCU and LEDs power supply

The schematic for this design is shown in figure 10 of appendix E. We use a VPS12-6300 transformer to step down the wall voltage to 6.3 V AC RMS. This results in a peak AC voltage of 8.9 V AC. The output of the transformer then fed into a robust DFB2060 bridge rectifier, which can handle currents up to 20 A, but has a forward voltage drop of 1.1 V. this means that our theoretical DC output voltage would be 7.8

V. actual output voltage was measured to be 7.27 V under load. this difference could be due to internal variances in the bridge rectifier or the transformer. Equation (1) was used to determine an appropriate filter capacitance value. The value for i was chosen to be 10 A. this was determined by adding up each LED's maximum current ratings at the the supplied voltage. the value of t is 8.3ms (milliseconds), which is the half cycle time for 60 Hz AC. ΔV was arbitrarily chosen at 1V and was subject to change after experimentation. Using equation (1) and the parameter values listed above, the necessary filter capacitor value was determined to 83,000 uF (microfarads). an 82,000 uF capacitor was purchased for the filter capacitor because of availability. After experimentation, the three color LED's were each run at 350 mA (milliamps), and the three white LEDs were each run at 700 mA. with the lowered current draw to 3.15 A. The values for capacitance and half cycle time were kept constant, and the output voltage ripple of the LED power supply, as determined by equation (1), was then determined to be 0.32 V.

The LF33CDT-TRY 3.3 V DC linear regulator was then connected to the LED supply voltage line in order to power the MCU, as shown in figure 11 in appendix E. input and output filter capacitance values were chosen to be 0.1 uF and 2.2 uF respectively. these values were chosen by recommendation of the linear regulator's data sheet.

2.3.1.2 Induction power supply

The schematic for this design is shown in figure 10 of appendix E. For the induction circuit power supply, we used a 500 VA (Volt-Amp) rated VPT48-10400 transformer to step down the wall voltage to 24 V AC RMS. 500 VA was chosen for robustness as we were unsure how much power the induction system would require. the 24 V from the secondary of the transformer has a peak voltage of $24 * \sqrt{2} = 33.8$V AC The output of the transformer then fed into a robust DFB2060 bridge rectifier, which can handle currents up to 20 A, but has a forward voltage drop of 1.1 V. This means that our theoretical DC output voltage would be 32.8 V. Actual output voltage was measured to be 31.5 V DC under load. Using Equation 1, we arrive at our desired filter capacitance value. The maximum current was estimated to be roughly 10 A. This assumption is based off a previous experiment where our induction circuit drew roughly 6 A at full load with a 20 V supply voltage. The value of t is 8.3ms as above in section 2.3.1.1 because this power supply also has 60 Hz input voltage. We chose an arbitrary ripple allowance of 2 V. Having a ripple of around 6% in the DC voltage input would not adversely affect the induction circuit [8]. Using equation (1) and the values listed above, the minimum filter capacitance value was chosen to be 41,500 uF. A 47,000 uF capacitor was chosen due to availability.

2.3.2 Microcontroller Unit

The MCU outputs to the LEDs are four PWM signals controlled to change the brightness of each of the four colors. The MOSFET signal gets high or low depending on the temperature sensor signal, as shown in Table 1.

Table 1: On and Off output of the MCU relationship with temperature sensor input

Temperature (°C)	**Temp. Sensor Voltage (V)**	**MCU heat output to MOSFET (V)**

>50°C	>1.0	0
<=50°C	<=1.0	5

In the Appendix E there is also a copy of the code uploaded to the MCU and a table with the different inputs and outputs.

2.3.3 LED Circuit

Figure 12 Appendix E details the circuit schematic for the LED Light System. This is physically constructed on the LED PCB shown in Figure 8 Appendix C. To calculate our theoretical brightness, we measure 3 single-die white LEDs that produce a total of 900 lumen at our operating current of 700 mA. These LEDs operate at a viewing angle of 125∘according to the data sheet [6]. Using equation (2):

$$\text{beam width} = 2*\text{distance}*tan(\frac{viewing\ angle}{2}) = 2*0.3*tan(\frac{125}{2}) = 1.15\text{ m} \tag{2}$$

we get beam width = 1.15 m. Using equation 3:

$$\text{coverage area} = \pi(\frac{beam\ width}{2})^2 = \pi(\frac{1.15}{2})^2 = 1.04\ m^2 \tag{3}$$

our coverage area at one foot away is 1.04 m^2. This gives us a total illuminance of:

$$\text{lux} = \text{lumens}/m^2 = 900/1.04 = 865\text{ lux}$$

To calculate the value of the current protection resistors, we experimentally measured the voltage across each diode and divided by the typical current of 350 mA for RGB and 700 mA for white. Using values from Appendix D, the resistances for each color are calculated using equation (4)-(7):

$$R_{RED} = \frac{V}{I} = \frac{Supply\ Voltage - Voltage\ Drop}{Typical\ Current(A)} = \frac{6.93 - 1.69}{0.350} = 15.0\ \Omega \tag{4}$$

$$R_{GREEN} = \frac{6.93 - 2.33}{0.350} = 13.1\ \Omega \tag{5}$$

$$R_{BLUE} = \frac{6.93 - 2.49}{0.350} = 12.7\ \Omega \tag{6}$$

$$R_{WHITE} = \frac{7.27 - 3.11}{0.7} = 5.9\ \Omega \tag{7}$$

Therefore our current protection resistance requirements for RGBW are 15.0 Ω, 13.1 Ω, 12.7 Ω, and 5.9 Ω, respectively. Equations (8)-(11) below show us that our power dissipation requirements are 0.60 W, 0.82 W, 0.87 W, and 2.18 W.

$$P_{RED} = Voltage\ Drop(V) * Typical\ Current(A) = 1.69*0.350 = 0.60\text{ W} \tag{8}$$

$$P_{GREEN} = 2.33*0.350 = 0.82\text{ W} \tag{9}$$

$$P_{BLUE} = 2.49*0.350 = 0.87\text{ W} \tag{10}$$

$$P_{WHITE} = 3.11 * 0.700 = 2.18 \text{ W} \quad (11)$$

2.3.4 Induction System (ZVS Driver)

The ZVS driver circuit is shown in figure 13 of appendix E. To build a ZVS driver of our own, we first determined the suitable input DC voltage. we purchased a pre-built zvs driver, shown in figure 7 of appendix B, to experiment with. The highest voltage we could experiment with was 20 V. At 20 V under no load, our copper coil drew 3A. Using equation (12), we determined that 60 W (watts) were consumed. We consider this wasted power.

$$P = V I \quad (12)$$

To determine minimum wattage needed to be induced in the lava lamp, we used equation [13] below. The lava lamp was determined to have roughly 0.75 litres, the temp change was determined to be 47 degrees fahrenheit, and heat up time was 10 minutes (or ⅙ of an hour) [8].

$$Watts = 0.78 * Litres * \Delta T \ (in\ °F) \ / \ Heat - Up\ Time\ (in\ hrs) \quad (13)$$

Using the above values, the power required to heat the water to operational temperature within 10 minutes was determined to be roughly 164 watts. Using the pre made induction circuit supplied at 20 V, a workpiece load was tested and we recorded that the current jumped from 3 A to 7 A. With the above power equation, using the difference in current from when the workpiece load was added, we estimated that 80W was induced in the water. Based on transformer price and availability, we selected a 115:24 turn transformer that, when rectified, would produce 31.5 V DC under load. we estimated that this power supply would produce the required wattage induced in the workpiece.

For the construction of our own ZVS driver to heat the lamp, the input MOSFET gate resistors were set at 470 ohms in order to limit inrush current into the gate pin. Too much current inrush could harm the device. 10k resistors were chosen for the pulldown resistors to prevent MOSFET latchup. 12 V zener diodes were chosen to limit the gate voltage, and to set minimum voltage for operation. The final components, the inductor and the capacitor, have to be chosen carefully to determine the oscillation frequency, which is determined by equation (14).

$$f = 1 \,/\, (2\pi * \sqrt{[L * C]}) \quad (14)$$

In this equation, f is oscillation frequency in hertz, L is input inductance in henries, and C is the capacitor value in Farads. the input inductor was chosen to be 100 uH (microhenries) and the capacitor was chosen to be 440 nF (nanofarads). This combination of values produces an oscillation frequency of 24 kHz (kilohertz). The capacitors were arranged in a parallel network of two 220 nF capacitors in order to split the current load between them [9]. Oscillation frequency needs to be high enough to surpass the critical frequency of the workpiece. Efficiency is highest at the critical frequency, and drops off slowly as the frequency increases[10]. Knowing this, we chose an arbitrarily high oscillation frequency of 24 kHz. The resulting ZVS driver is shown below in figure 6 in appendix B.

2.3.5 Temperature Sensors

There is a linear relationship between the output voltage and the temperature read by the temperature sensor (Equation 15). This information is used to regulate the output of the induction circuit and make sure that the lamp doesn't get too hot.

$$Voltage[V] = 0.75 + (Temperature[°C] \ - 25) * 0.01 \quad (15)$$

3. Design Verification

3.1 Power

Both power systems successfully completed their requirements. The MCU and LED power supply was fully able to power the entire PCB system, and the induction power supply was fully able to power the entire induction system.

3.1.1 MCU and LEDs power supply

The MCU and LED power supply was able to take the input 120V AC from the wall and successfully convert it to 7.27 V DC, as read by a digital multimeter. This 7.27 V source was able to successfully power the LED. The 7.27 V LED supply voltage was then successfully stepped down to 3.3 V DC to power the MCU. This was also measured by a digital multimeter. This reading is shown in figure 14 of appendix E.

We have a picture of a no-load voltage output test for the LED power supply, but not for a loaded test. At no load, the power converter outputs 8.6 V DC, but that voltage level is lowered to 7.3 V DC under load. The no-load voltage reading is shown in figure 15 of appendix E.

3.1.2 Induction power supply

The Induction power supply was able to take the input 120 V AC from the wall and successfully convert it to 31.5 V DC, as read by a digital multimeter. It was also robust enough to provide as much power as the induction system required. We have a picture of a no-load voltage output test for the LED power supply, but not for a loaded test. At no load, the power converter outputs 34.7 V DC, but that voltage level is lowered to 31.5 V DC under load. The no-load voltage reading is shown in figure 16 of appendix E.

3.2 Control

3.2.1 Microcontroller Unit

The MCU fulfilled all the requirements, since it was able to toggle through all seven different colors and five brightness modes for each color, see Figure 3. The control of the induction power supply was also functional: whenever the temperature sensor input went over 1 V (50 °C) the mosfet was turned off.

Figure 3: Demonstration of the 7 different colors switching.

3.3 Lighting

3.3.1 LED Light System

According to our prior calculations in Section 2.3.3, we should not have been able to meet our brightness requirement of 3000 lux at 1 ft. However, we were able to verify our light source producing 3566 lux in practice (Figure 19 Appendix F). This is either due to the fact that the light sensor on our Samsung Galaxy S6 phone was not accurate, or that the viewing angle of the LEDs were much narrower than specified on the data sheet. For red, orange, yellow, green, blue, and violet we obtained illuminances of 1124, 1149, 1844, 1574, 1119, and 2489 lux at 1 ft respectively (Figure 18 Appendix F).

We successfully met our other requirements for switching between five brightnesses and seven colors by observing brightness and color changes in the final version of the lava lamp.

3.4 Induction

The induction system failed to meet its requirements. The active coil we chose was insufficient for this project, and we believe it was the limiting factor for all three induction requirements. One requirement is that the active coil remain under 44 °C after extended operation. The copper coil we used reached upwards of 125 °C. The second requirement is for the water to be heated to 50 °C within 10 minutes. Because of the coil's excessive temperature, we were unable to run a full 10 minute test under 31.5 V input voltage to the ZVS driver due to fear of overheating. The ZVS driver was run at 20 V for 10 minutes and the water was only able to be heated to 40 °C. The test had to be stopped at the 10 minute mark because the coil's enclosure began to melt. A chart showing the progressive heating of the water over time for this test is shown in figure 17 in appendix E. Because we were unable to heat the water to rated temperature, we were unable to perform the final verification test for the glass surface temperature remaining under 44 °C.

Initial testing of the hand-built ZVS driver resulted in burning out of the MOSFETs, so all future induction testing was done with the prebuilt ZVS driver. After completion of the hand-built ZVS driver, we realized that the pre-built driver had a very similar design (we purchased it without any schematic info), and were able to use our understanding of the ZVS switching driver to understand and operate the pre-built circuit.

3.5 Measurement

3.5.1 Temperature sensors

The temperature sensors met all requirements needed for the design. They were tested to:

- Give accurate temperature every 15 seconds.
- Operate in a range of voltages readable by the MCU: 0.1 to 1.75 V (-40 to 125 °C).
- Have a precision of $\pm$2 °C.

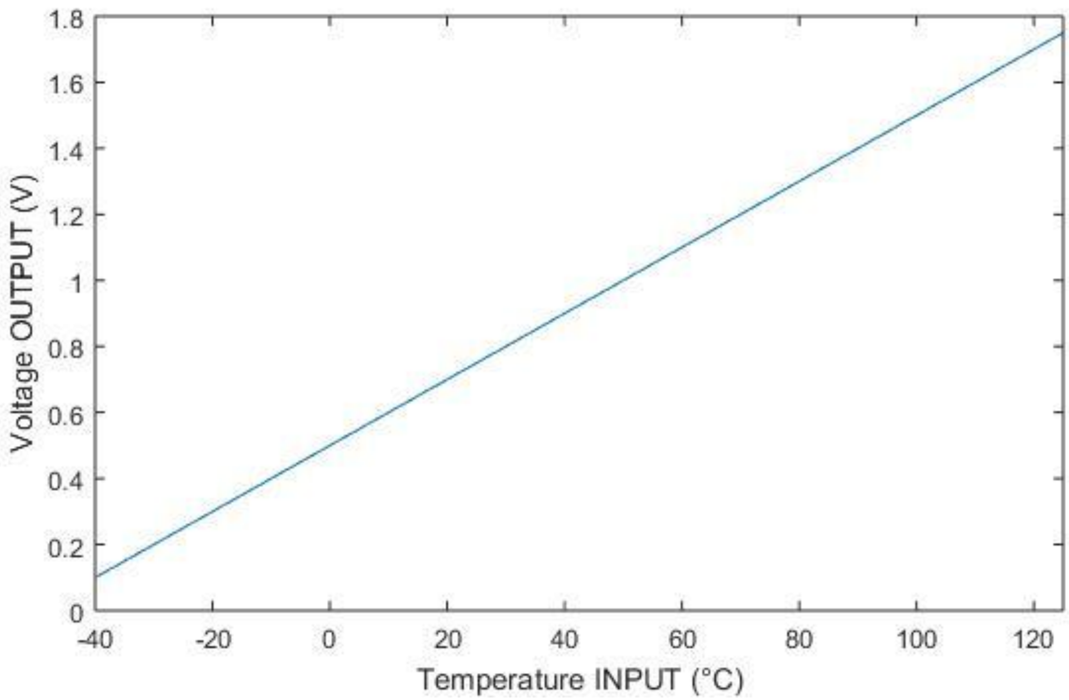

Figure 4: Graph of the relationship between the input and the output of the temperature sensor

4. Costs

4.1 Parts

Table 2: Parts Costs

Part	Function	Unit Price	Amount	Total Price
ATMEGA328P-PU	MCU	$3.37	1	$3.37
PSMN022-30PL	n-MOSFET for LEDs	$0.58	4	$2.32
RGBW CREE LED's	Color LED	$7.07	3	$21.21
White CREE LED's	White LED	$6.79	3	$20.37
13 Ohm 2 W resistors	Current protection	$0.32	9	$2.99
6.8 Ohm 4W resistors	Current protection	$1.05	3	$3.15
PCBs	PCBWay	-	-	$33.00
TMP36	Temperature sensor	$1.50	2	$3.00
Tactile button	User input	$0.11	2	$0.22
Assorted washers	Induction workpiece	$13.99	2	$13.99
Glass beaker	Induction testing	$15.95	1	$15.95
VPT48-10400	115:24 transformer	$91.87	1	$91.87
DFB2060	20A bridge rectifier	$1.61	2	$3.22
ESMH400VND473MB80T	Filter capacitor	$10.80	2	$21.60
VPS12-6300	115:6.3 transformer	$20.71	1	$20.71
513102B02500G	Heat sinks	$0.98	2	$1.96
LF33CDT-TRY	3.3V linear regulator	$0.84	1	$0.84
IPP17N25S3-100	n-MOSFET for induction control	$1.17	2	$1.17
B64290L58X830	Ferrite toroid	$2.75	2	$5.50
1000W pre-built induction heater	Testing	$38.99	1	$38.99
IPI600N25N G	ZVS Driver MOSFET	$2.36	2	$4.72
Double sided tape	Coil winding	$8.16	1	$8.16
Jumper wires	PCB connections	$7.99	1	$7.99
Small Diodes/Capacitors/Resistors	Miscellaneous	-	-	$10.00
Total				**$336.21**

4.2 Labor

Table 2 shows the breakdown of labor costs for this project. At an hourly rate of $ 30 for each group member the total cost for the course of this semester would be $ 42,215.

Table 3: Labor costs

Personnel	
Assumed Starting Salary	$70,000
Per Hour Rate	$30
Total Hours (20 per week)(12 total weeks)	240
Personnel Cost (for 3 members)	$21,600
Total Cost (12 week salary + personal cost)	**$42,215**

5. Conclusion

5.1 Accomplishments

Our primary goal was to heat a lava lamp by means of induction heating. While we didn't manage to successfully operate a lamp with induction heating, we managed to prove that the concept is viable. With more work, induction can be used as a lava lamp heat source.

We, however, did completely accomplish the task of creating a significantly brighter lava lamp with user control. Our array of three white LEDs and three color LEDs are much brighter than the previous design and allowed the lava lamp to function as a viable light source. The control system also successfully allowed the user to cycle between seven colors and five brightness settings.

We were also able to successfully design power supplies to power the LED system, the MCU system, and the induction system. The induction power supply was designed independently from the LED and MCU power supply, which allowed for the induction heat source to be turned off while the lights and MCU control remained active.

5.2 Uncertainties

Construction of the active coil of the induction system left uncertainties in the design of the system. All designs that we tried became too hot when operated at 31.5 V, which is due to the fact that the coils we constructed had too high of a series resistance. We expect that litz wire, which is especially designed to reduce the skin effect and carry alternating current with minimal losses, would allow the lamp to be heated to rated temperature within ten minutes. We tried a coil made of litz wire, but its gauge was much too small and caused the insulation to burn up, which in turn caused a short circuit and blew out the power supply. The next step on the induction heating of a lava lamp should be trying a pancake coil made of 9 gauge litz wire and testing the requirements of fast heating, low coil temperature, and low glass temperature.

5.3 Ethical considerations

Safety is a big concern for us and is at the forefront of our project requirements. When designing a lava lamp with a new inductive heating circuit we must be mindful of electrical, chemical, and thermal hazards.

When developing our system we followed certain safety procedures to avoid danger of electric shock. We worked at a lab bench when testing our circuit at all times. Before adjusting any circuit we will made sure to unplug all power sources. We made sure our circuits were properly grounded at all times. Finally, we tested our power system with an oscilloscope before connecting to our control and energy systems.

The composition of the wax inside of our lava globe presents a chemical hazard. While the chemical formula of the LAVALITE® MOTION LAMP is a trade secret, the official US Patent for lava lamps states that the wax contains a chemical called carbon tetrachloride[11]. Carbon tetrachloride causes eye, skin, ingestion, and respiratory irritation so we must wear gloves when handling the wax at all times and avoid ingestion[12]. If the wax inside is replaced with our own formula we must adhere to proper safety procedures for any chemicals involved according to the OSHA guidelines.

The induction system and heated glass also presents a thermal hazard, as well as a physical hazard from the risk of explosion. Temperatures of greater than 118 °F on the external surface of the lamp cause burns when touched, and excessively high temperatures on the bottom surface the glass may cause it to fracture or even explode. This is why our design features a temperature feedback system to keep the external glass below this temperature, keeping users safe. To ensure safety for the designers, preliminary thermal tests were run on the power system to make sure internal components did not overheat. Then, the power system was be joined to the temperature system and additional testing would be performed to guarantee proper feedback before heating the lava globe. If at any time during testing the temperature feedback failed or rose above the desired temperature, a group member was ready to manually disconnect power from the induction system.

5.4 Future work

Although great strides have been made during this semester with regards to the lava lamp, there is still more that can be done. First and foremost, we would like to improve on our induction design. We have successfully proven induction to be a viable method of heating a lava lamp. In the future, we would like to replace the copper tube active coil with a litz wire coil. This would allow the induction heater to operate at a higher voltage and heat the water faster.

We would also like to work on integration of all components. We were unable to connect all subcomponents into a single working lava lamp unit. We would like to see all components working together and have control signals operating everything autonomously.

The final improvement would be creating a more aesthetic physical design. This goes along with integration. Once all components are integrated in a lamp base, work can be done to make it attractive as a consumer product.

References

[1] "The Physics Of A Lava Lamp," *A Moment of Science - Indiana Public Media*. [Online]. Available: http://indianapublicmedia.org/amomentofscience/physics-lava-lamp/. [Accessed: 03-May-2017].

[2] W. Lighting, "Lava lamp for kids," *Warisan Lighting*, 16-Jun-2016. [Online]. Available: http://warisanlighting.com/lava-lamp-for-kids.html. [Accessed: 03-May-2017].

[3] "Lava Lamp Sience." [Online]. Available: http://www.slcschools.org/departments/curriculum/science/grade-3-to-6/grade-6/documents/s6-o1-lesson-lava-lamp-science.pdf. [Accessed: 03-May-2017].

[4] "Fire Dynamics," *NIST*. [Online]. Available: https://www.nist.gov/%3Cfront%3E/fire-dynamics. [Accessed: 03-May-2017].

[5] "Cree® XLamp® XM-L Color LEDs Product Family Data Sheet," *CREE*. [Online]. Available: http://www.cree.com/led-components/media/documents/XLampXML_Color-5A.pdf. [Accessed: 05-May-2017].

[6] "Cree® XLamp® XM-L LEDs Product Family Data Sheet," *CREE*. [Online]. Available: http://www.cree.com/led-components/media/documents/XLampXML-11E.pdf. [Accessed: 05-May-2017].

[7] J. Keith, "Capacitor Input Filter Calculation." [Online]. Available: http://www.electroschematics.com/7048/capacitor-input-filter-calculation/. [Accessed: 05-May-2017].

[8] I. Renwick, "How Much Wattage do I Need? | Tutco-Farnam Custom Products," *Tutco-Farnam Custom Products*, 01-Sep-2012. [Online]. Available: https://farnam-custom.com/resources/engineer-talk/how-much-wattage-do-i-need/. [Accessed: 05-May-2017].

[9] "ZVS (Mazilli) Driver | Adam Munich." [Online]. Available: http://adammunich.com/zvs-driver/. [Accessed: 05-May-2017].

[10] "Theory of heating by induction," *Elegron*. [Online]. Available: http://www.elegron.pl/files/TheoryHeating.pdf. [Accessed: 05-May-2017].

[11] W. E. C, "Display device," 3570156, 16-Mar-1971.

[12] "Material Safety Data Sheet. Carbon tetrachloride." [Online]. Available: http://fscimage.fishersci.com/msds/90116.htm. [Accessed: 05-May-2017].

Appendix A: Requirement and Verification Table

Table 4: System requirements and verifications

Requirement	Verification	Verification Status
MCU and LEDs power supply (5 points) 1. Accept 120 V AC input from wall outlet and be able to supply stable 8.6 V DC to LED circuit 2. Accept 120 V AC input from wall outlet and be able to supply stable 3.3 V DC to MCU unit	**MCU and LEDs power supply (5 points)** 1. Verification a. Connect power to wall. b. Use a multimeter to measure the voltage across rectifier output. 2. Verification a. Connect power to wall. b. Use a multimeter to measure the voltage across rectifier output.	Y Y
Induction Power Supply (5 points) 3. Accept 120 V AC input from wall outlet and be able to supply stable 32 V (+-1 V) DC to Induction Circuit	**Induction Power Supply (5 points)** 3. Verification a. Connect power supply to wall. b. Connect induction circuit to power supply. c. Use a multimeter to ensure rectifier output is 32 VDC (±1 V).	Y
Microcontroller Unit (10 points) 4. Able to control LED lights. 5. Receive temperature sensor data (voltage) and interpret it, powering the induction circuit on and off.	**Microcontroller Unit (10 points)** 4. Verification a. Connect LED circuit to MCU. b. Power circuit on, verify if working. 5. Verification a. Connect voltage supply to MCU. b. Increase and decrease voltage, compare MCU voltage output. With under 1.0 V (50 °C) ON, with over 1.05 V (55 °C) OFF.	Y Y
LED Light Source (15 points) 6. Must produce at least 3000 lumens of light	**LED Light Source (15 points)** 6. Verification a. Turn the lamp to max	

7. Must be able to control illuminance at five equally spaced increments 8. Must be able to switch between red, orange, yellow, green, blue, purple, and ultra-bright white lighting	brightness and use a digital light meter at the glass surface of the lava globe. b. Measure the max setting to be least 3000 lux in any one direction. 7. Verification a. Press on-unit input to cycle through five brightness settings. b. Measure lux at 20%, 40%, 60%, 80%, and 100% of the maximum recorded setting with error of <50 lux. 8. Verification a. Press on-unit input to cycle between seven color modes. b. Take a cell phone picture of the lamp in a dark room c. Upload image to imagecolorpicker.com and measure the color of the brightest part of the image d. Digital RGB value must be within 50 for every parameter of the following codes: i. Red: (255,0,0) ii. Orange:(255,75, 0) iii. Yellow: (255,255,0) iv. Green: (0,255,0) v. Blue:(0,0,255) vi. Violet: (255,0,255)	Y Y Y
Induction (10 points) 9. Heat up 0.75 litres of liquid from room temperature to 50 °C within 10 minutes	**Induction (10 points)** 9. Verification a. Induction circuit to a power supply at 20 V -	

10. Make sure active coil does not exceed 44 °C 11. Ensure surface temperature of glass does not exceed 44 °C	32 V. b. Confirm that loaded operation induces at least 80 W into the water (using input DC voltage multiplied with change in current when load is applied) c. Confirm that coil doesn't start melting or start burning. d. Using a thermometer, measure the water temperature and verify it reaches 50 °C within the 10 minute window. 10. Verification a. Perform verification test number 6. b. After 10 minutes, remove the water container and measure coil temperature with a thermometer. 11. Verification a. Perform verification test number 6. b. Measure bottom inch on the side surface of glass with thermometer and verify measurement does not exceed 44 °C	N N N
Temperature Sensors (5 points) 12. Measurements once every 15 seconds	**Temperature Sensors (5 points)** 12. Verification a. Attach the temperature sensor to a warm object. b. Separate the temperature sensor and leave it on ambient temperature, connected to a measurement device to track the heat change. c. If the temperature matches the independent	Y

13. Analog output of voltage readable by MCU. 14. Precision of ±2 °C	thermometer after 15 seconds then the requirement is met. 13. Verification a. Use a multimeter to ensure that voltage reading from sensor doesn’t exceed for all operational temperatures. 14. Verification a. Measure different temperatures. Verify sensor output, must be less than 20 mV(2 °C) difference than calculated.	Y Y

Appendix B: Physical Design

In Figure 5 is a sketch of what our physical design looked like. Induction heating allows us to achieve a faster and more energy efficient heat transfer, as the induction coil and workpiece are held close to each other. This also leaves space around them to place our LED lights, and in the bottom of the glass we can measure the temperature, which is be correlated to the current flowing through our coil, and thereby controlled by the MCU.

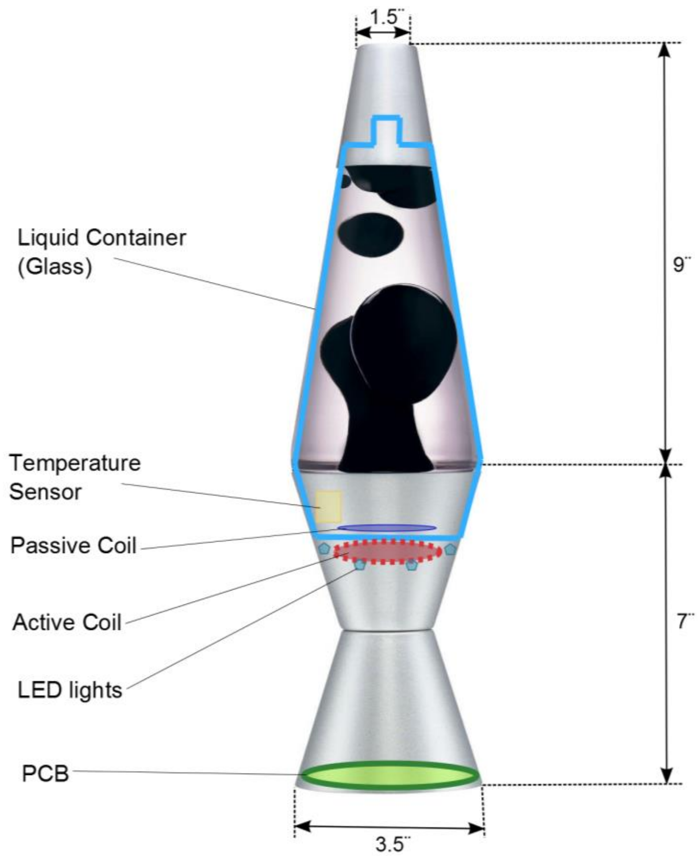

Figure 5: Physical design of the lava lamp

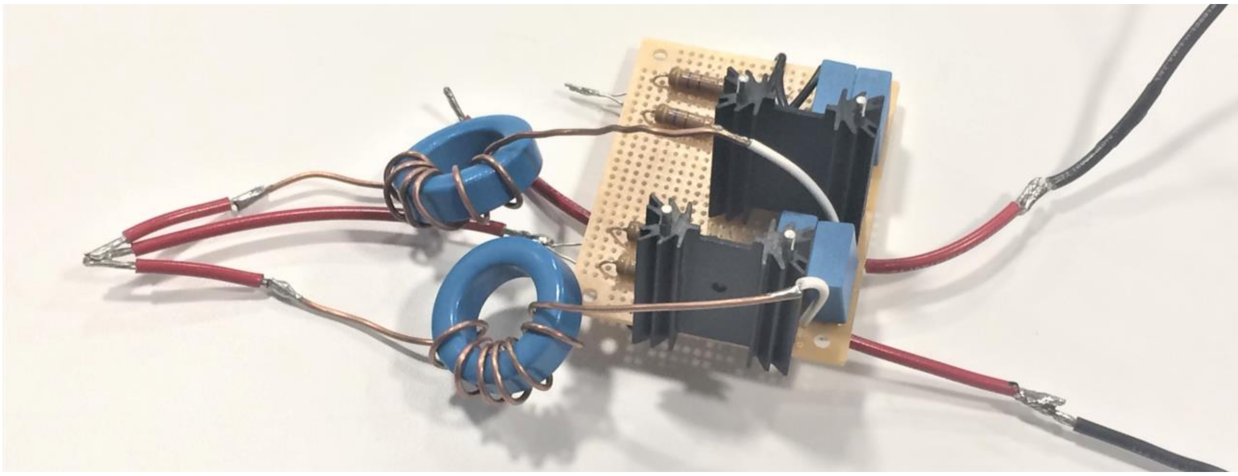

Figure 6: Hand built ZVS driver

Figure 7: Pre-built ZVS driver and copper tube coil

Appendix C: Assembled PCB

Both our MCU and LED circuits, as well as the circuit which powers them, were mounted on circular Printed Circuit Boards in order to fit them in the base. The result is shown in Figure X.

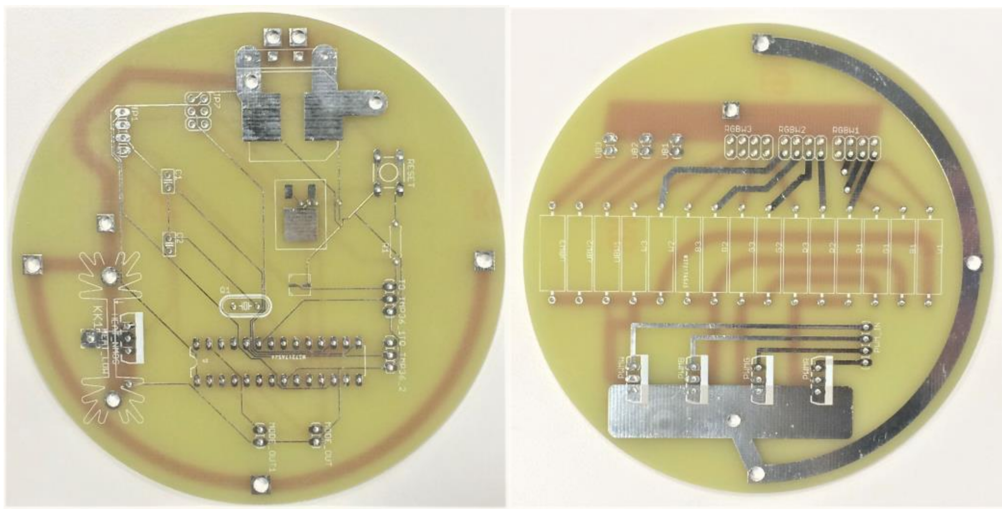

Figure 8: Printed Circuit Boards of the MCU and power (left) and LED (right) circuits

Appendix D: Tables

Table 5: RGB LED Power Draw

LED	Resistor value	Supply Voltage	Voltage drop	Current (I=V/R)	Power draw
R1	13 Ω	6.93 V	1.69 V	403 mA	0.681 W
R2	13 Ω	6.93 V	1.69 V	403 mA	0.681 W
R3	13 Ω	6.93 V	1.69 V	403 mA	0.681 W
G1	13 Ω	6.93 V	2.33 V	354 mA	0.825 W
G2	13 Ω	6.93 V	2.23 V	362 mA	0.807 W
G3	13 Ω	6.93 V	2.26 V	359 mA	0.811 W
B1	13 Ω	6.93 V	2.49 V	342 mA	0.852 W
B2	13 Ω	6.93 V	2.48 V	342 mA	0.848 W
B3	13 Ω	6.93 V	2.48 V	342 mA	0.848 W

Table 5: Single-Die White LED Power Draw

LED	Resistor value	Supply Voltage	Voltage drop	Current (I = V/R)	Power draw

White 1	6.8 Ω	7.27 V	3.11 V	611 mA	1.90 W
White 2	6.8 Ω	7.27 V	3.18 V	601 mA	1.91 W
White 3	6.8 Ω	7.27 V	3.10 V	611 mA	1.89 W

Table 6: Total LED Power Draw

All on	5.11 A	12.73 W
White	1.8 A	5.7 W
R/G/B	1.03 A	2.55 W
O/Y/V	2.1 A	5 W

Appendix E: Circuit Schematic Figures

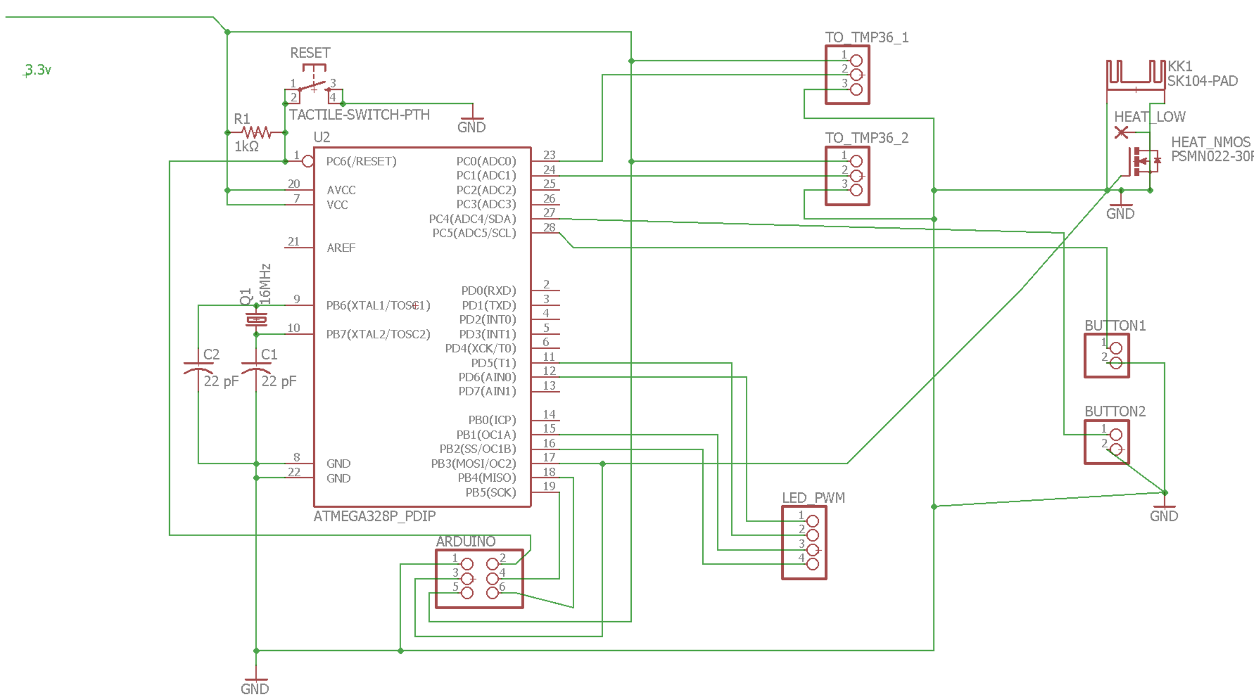

Figure 9: MCU schematic

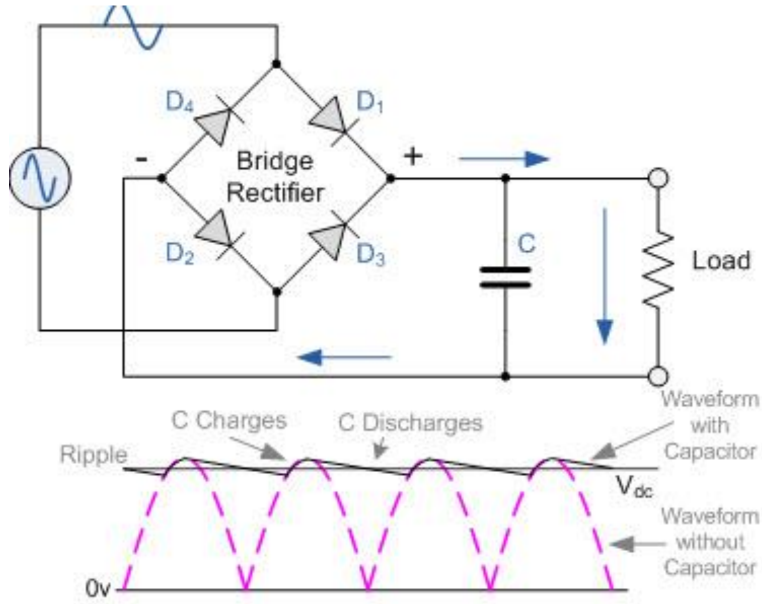

Figure 10: Full Wave Rectification

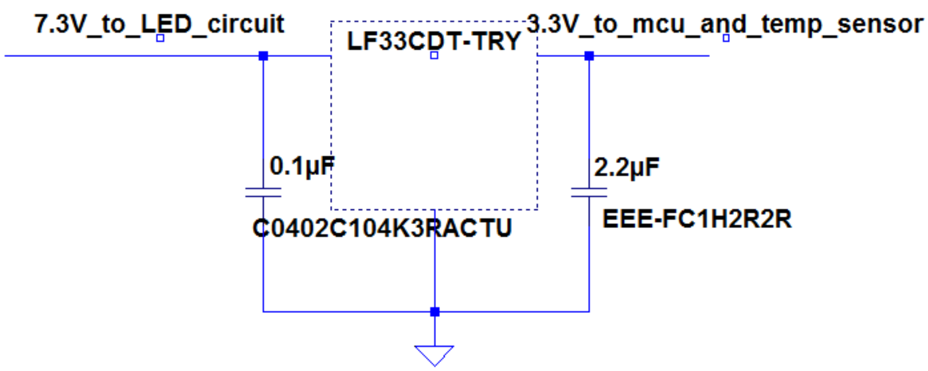

Figure 11: Linear Regulator Circuit Diagram

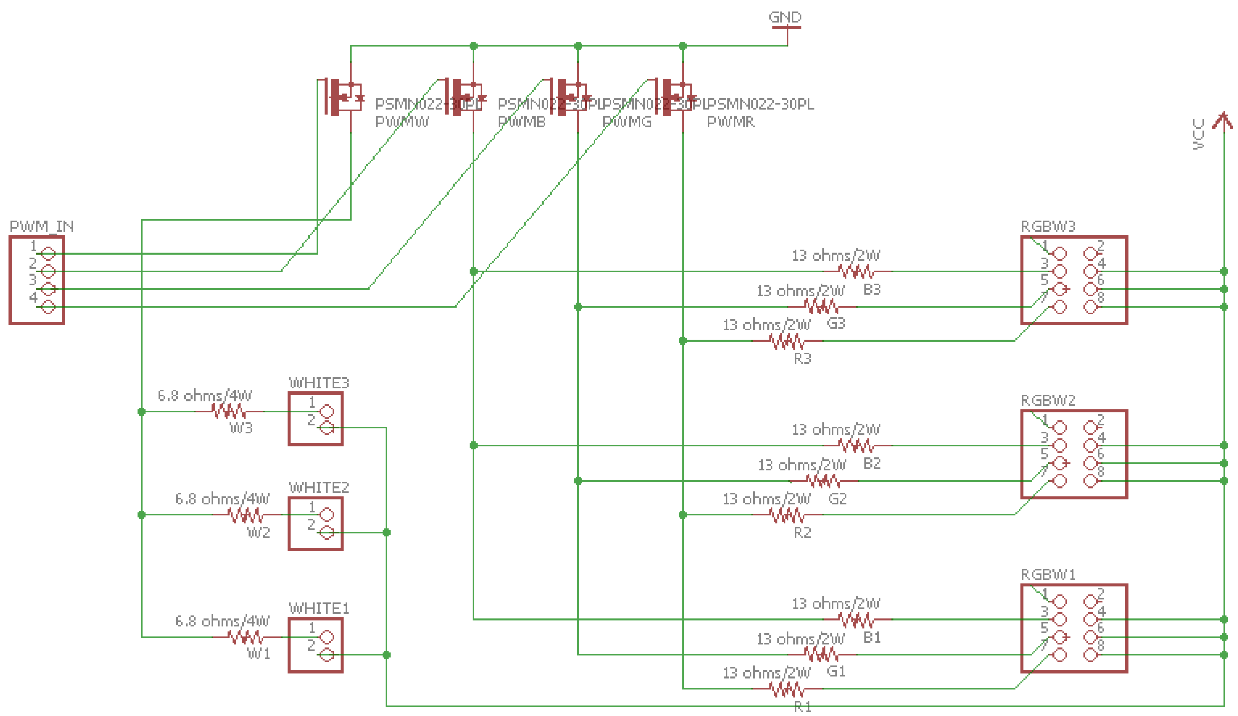

Figure 12: LED circuit schematic

An n-channel MOSFET controls the operation of each of the four RGBW color diodes. Each MOSFET's gate is connected to a PWM signal which is outputted from the MCU. Current protection resistors are connected in series with each color diode. Power is supplied by the PCB power circuit.

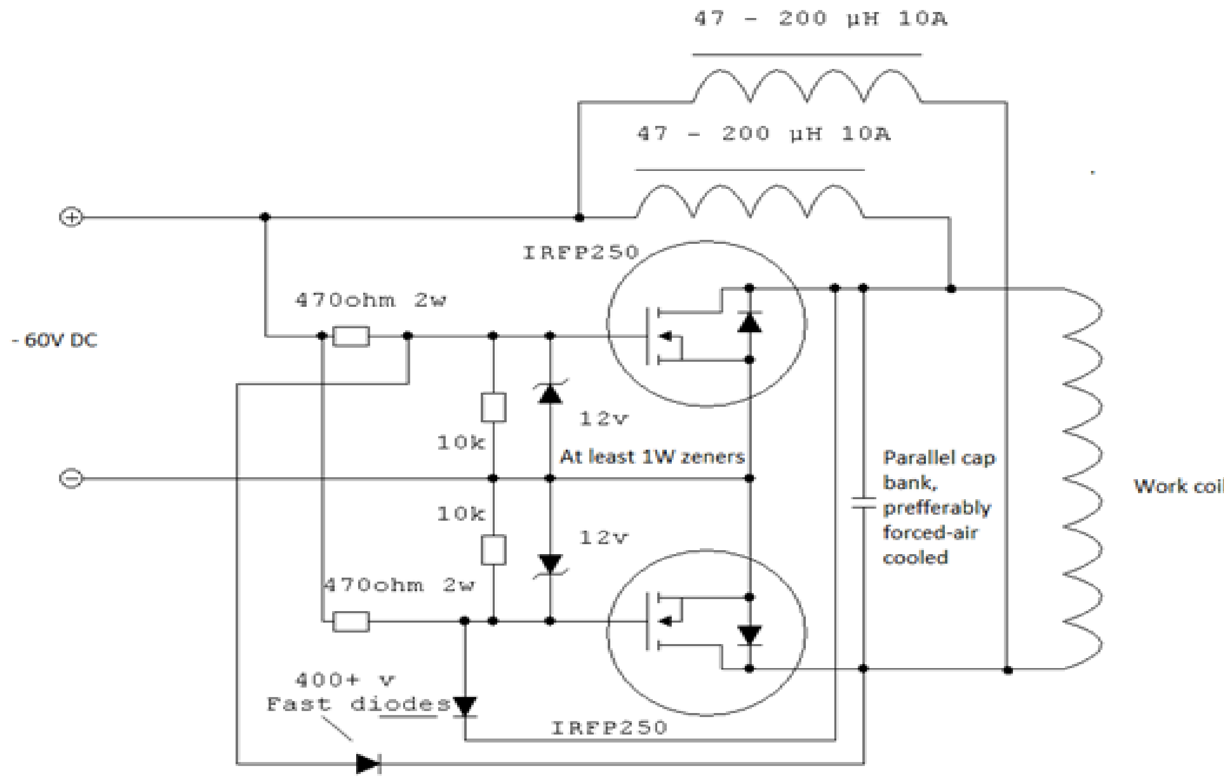

Figure 13: ZVS driver circuit schematic

Appendix F: Verification Figures

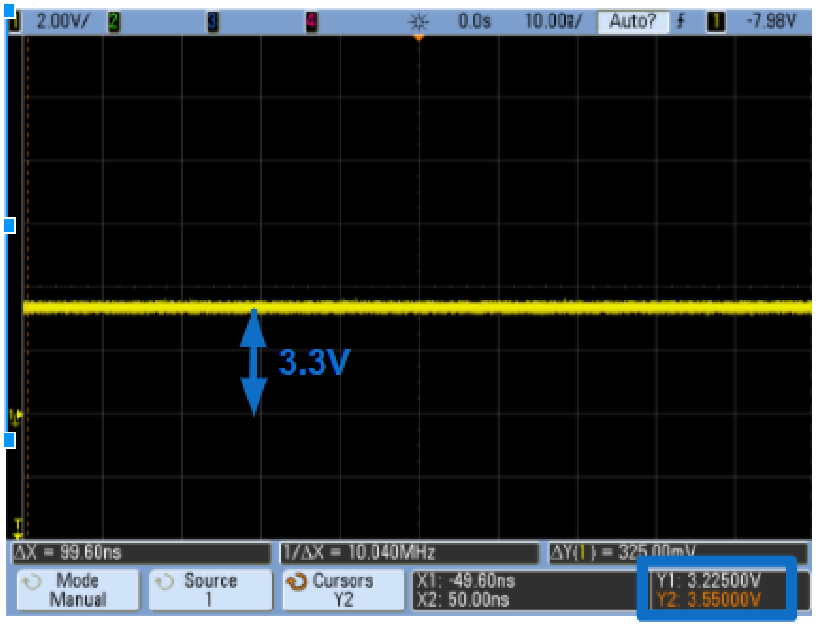

Figure 14: MCU output voltage verification

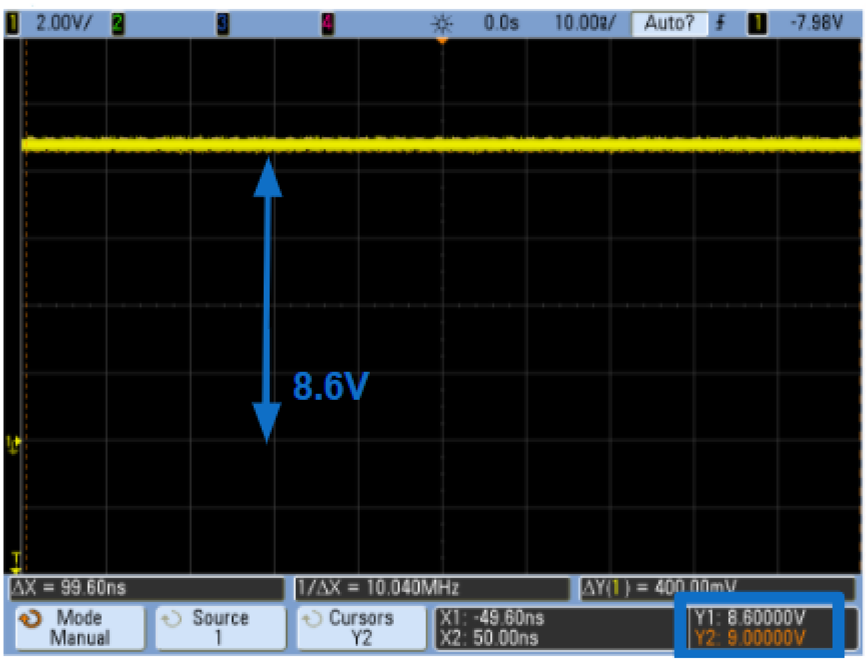

Figure 15: LED power supply output verification (no load)

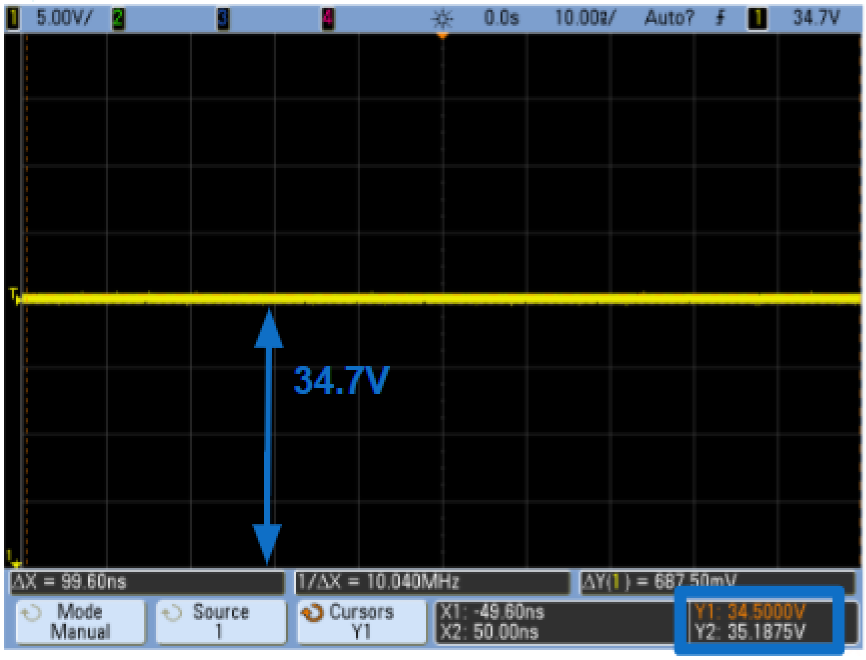

Figure 16: Induction power supply output verification (no load)

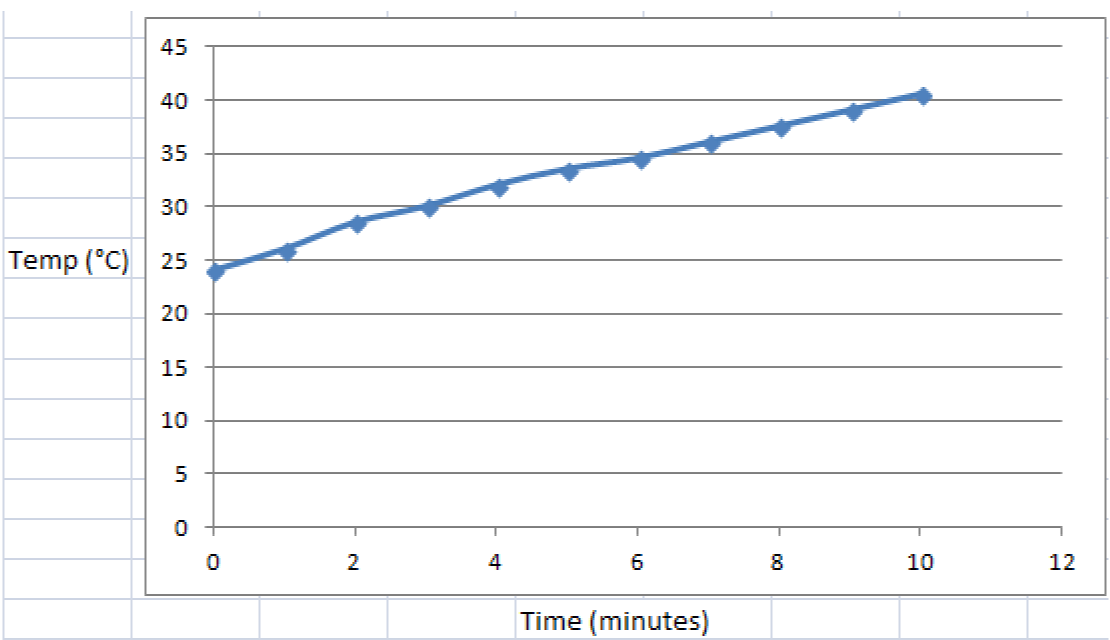

Figure 17: Water temperature vs time

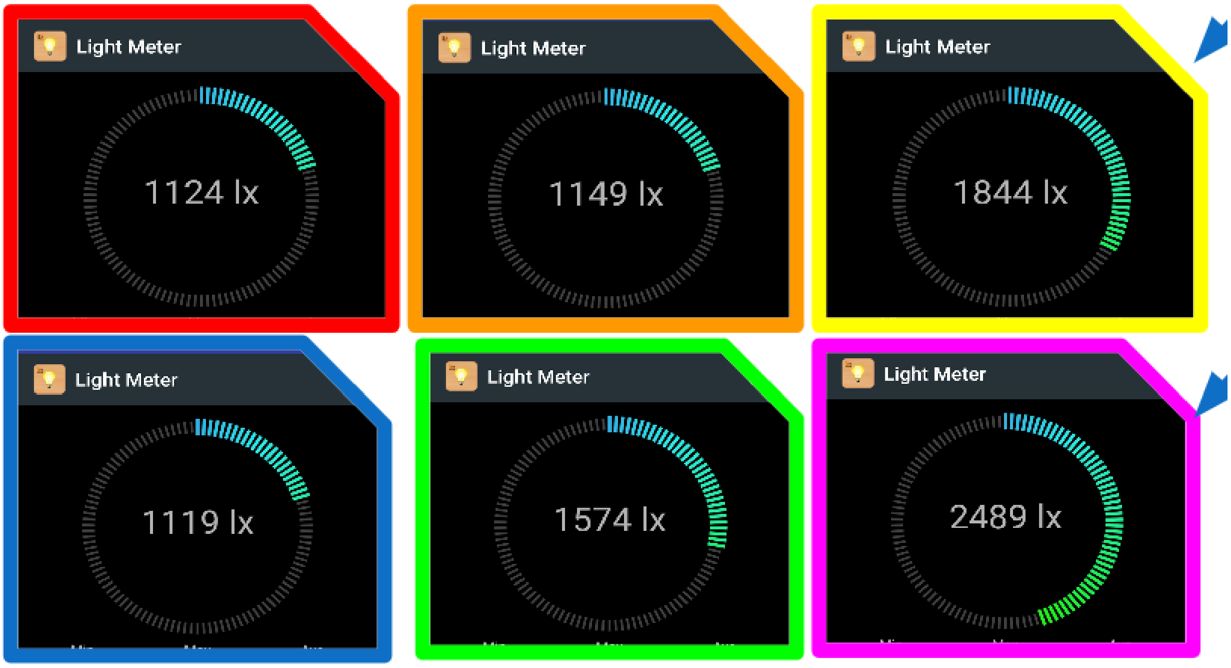

Figure 18: Color LED Illuminances at 1 ft

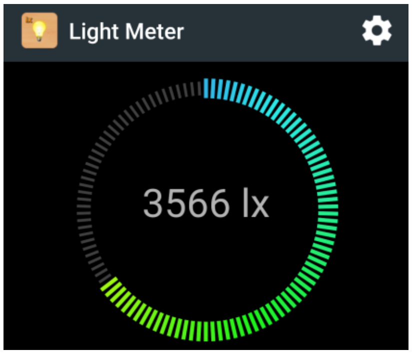

Figure 19: White LED Illuminance at 1 ft

Appendix G: Microcontroller Unit

To understand the functionality of the MCU, in Table 4 is the summary of the inputs and outputs, as well as a copy of the code used.

Table 6: Input and Output table for the MCU pin connections:

PIN	**TYPE**
Button 1	Input
Button 2	Input
Lava Temperature Sensor	Input
Heat Temperature Sensor	Input
Red LED PWM	Output
Green LED PWM	Output
Blue LED PWM	Output
White LED PWM	Output
Heat Output	Output

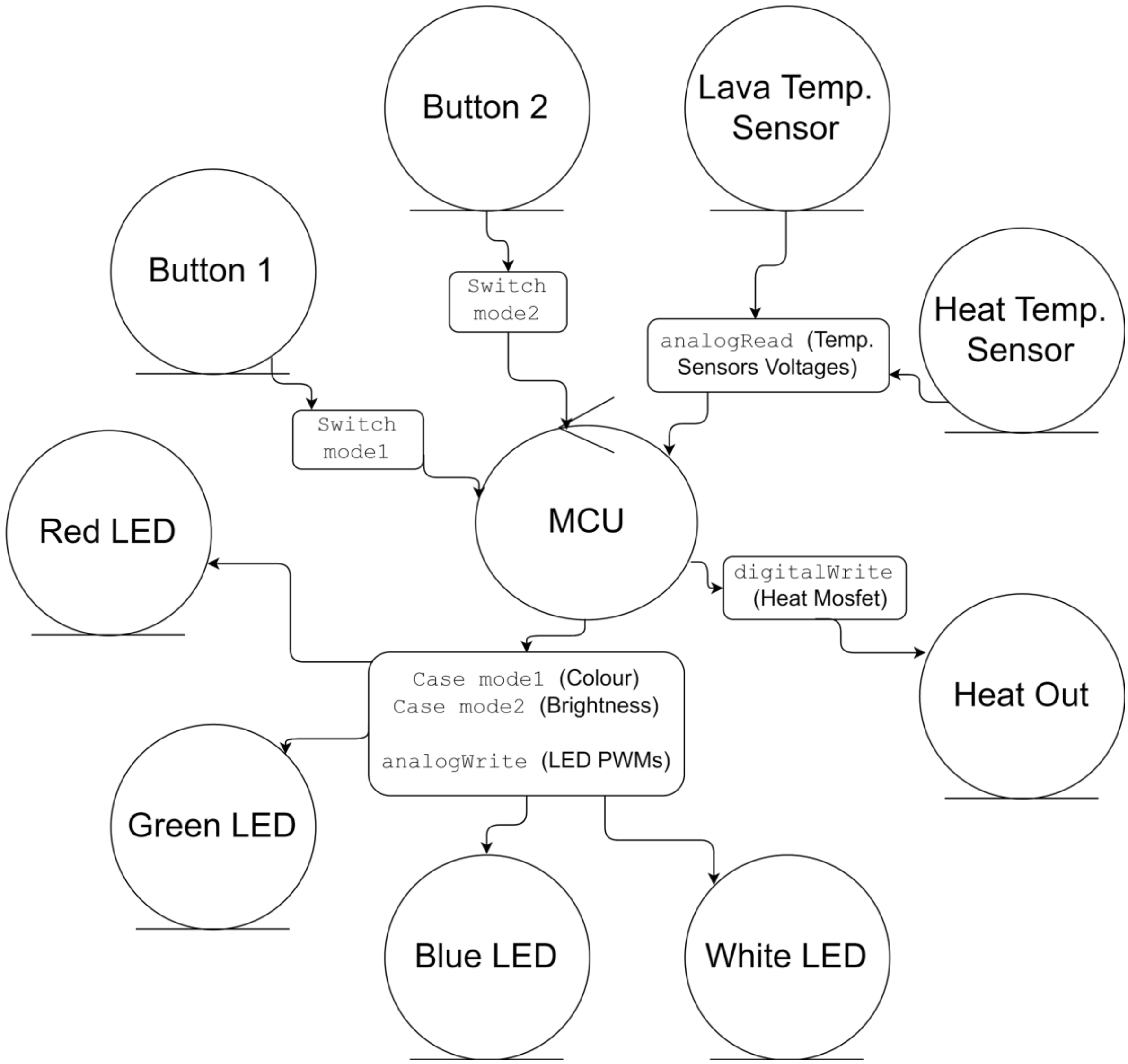

Figure 20: MCU flowchart

```
//Debug (uncomment the line below to enable serial monitor debugging
//#define DEBUG 1
////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                             Global Variables                           //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
//Pin Values
const int button1 = A4;  //PD2 (pin27) is button1 input
const int button2 = A5;  //PD2 (pin28) is button2 input
const int heat = 11;   //PB2 (pin17) is the heat PWM output
const int ledW = 10;    //PD3 (pin16) is LED4 (assume is W)
const int ledG = 5;    //PD5 (pin11) is LED2 (assume is G)
const int ledB = 9;    //PD6 (pin15) is LED3 (assume is B)
const int ledR = 6;    //PB1 (pin12) is LED1 (assume is R)
const int tempHeat=A0; //PC0 (pin23) is Heat Source Temp Sensor
const int tempLamp=A1; //PC1 (pin24) is Lava Lamp Temp Sensor
//Controller Parameters TODO Set These!!
float desiredTemp = 50.0;   //Desired Lava Lamp Tempin degrees Celsius
float maxHeatTemp = 70.0;   //Highest Allowable Heat Source Temp
float K = 10;              //Proportional gain constant    TOASK
```

```
float maxInHeat = 100;    //100 W
//Other Global Variables
double potVal = 0;
int button1Val = 0;
int oldbutton1Val = 0;
int button2Val = 0;
int oldbutton2Val = 0;
int mode1 = 0;     //mode1s 0 thru 4
int mode2 = 0;
int redVal = 0;
int greenVal = 0;
int blueVal = 0;
int whiteVal = 0;
int heatVal = 0;
float heatTempAvg = 25.0;  //heat source temp in Celsius. This is actually the average of the
last 8 temps using arrays below
float lampTempAvg = 25.0;  //lava lamp temp in Celsius. This is actually the average of the last
8 temps using arrays below
float heatTemp[8] = {25.0,25.0,25.0,25.0,25.0,25.0,25.0,25.0};   //arrays for last 8 temps to
average
float lampTemp[8] = {25.0,25.0,25.0,25.0,25.0,25.0,25.0,25.0};
float NEW=0;
float bye =0;
int heatIndex = 0;                                    //indices for the arrays for easy access
int lampIndex = 0;

///////////////////////////////////////////////////////////////////////////
//                                                                       //
//                             Initialize MCU                            //
//                                                                       //
///////////////////////////////////////////////////////////////////////////
void setup()
{
  #ifdef DEBUG
  Serial.begin(9600);
  #endif
  analogReference(EXTERNAL);
  pinMode(heat, OUTPUT);  //heat pwm output
  pinMode(ledG, OUTPUT);  //led pwm outputs
  pinMode(ledB, OUTPUT);
  pinMode(ledR, OUTPUT);
  pinMode(ledW, OUTPUT);
  pinMode(button1, INPUT); //button1 input
  pinMode(button2, INPUT); //button2 input
  pinMode(tempHeat, INPUT);
  pinMode(tempLamp, INPUT);
  Serial.begin(9600);
}
///////////////////////////////////////////////////////////////////////////
//                                                                       //
//                             Main Loop   (DONE)                        //
//                                                                       //
///////////////////////////////////////////////////////////////////////////
void loop()
{
  //Light
  userInput();  //Read User Input (button1, button2)
  calcLEDs();   //Calculate LED PWM Signals
  setLEDs();    //set LED PWM Signals;
  //Heat
  tempInput();  //Read Temperature Sensors
  calcHeat();   //Calculate Heat PWM Signals
  setHeat();    //Set Heat PWM Signals
}
///////////////////////////////////////////////////////////////////////////
//                                                                       //
//                              Functions                                //
//                                                                       //
///////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////////////
//                                                                       //
```

```
//                   Read User Input (button1, button2) DONE               //
//                                                                         //
/////////////////////////////////////////////////////////////////////////////
void userInput()
{
  //read the button1 value
  button1Val = digitalRead(button1);
  if(button1Val != oldbutton1Val)    //allow ourselves to switch
  {
     if(button1Val == HIGH)  //sense rising edge
     {
       mode1 = ++mode1 % 7;            //toggle to the next mode1
     }
     delay(200);
  }
  oldbutton1Val = button1Val;
  //read the button2 value
  button2Val = digitalRead(button2);
  if(button2Val != oldbutton2Val)    //allow ourselves to switch
  {
     if(button2Val == HIGH)  //sense rising edge
     {
       mode2 = ++mode2 % 5;            //toggle to the next mode2
     }
     delay(200);
  }
  oldbutton2Val = button2Val;
}
/////////////////////////////////////////////////////////////////////////////
//                                                                         //
//                    Calculate LED PWM Signals  DONE                      //
//                                                                         //
/////////////////////////////////////////////////////////////////////////////
void calcLEDs()
{
  //obtain the color first
  switch(mode1) {
      case 0:  //white
         #ifdef DEBUG
         Serial.print(" Red mode1  ");
         #endif
         redVal = 0;
         greenVal = 0;
         blueVal = 0;
         whiteVal = 255;
         break;
      case 1:  //red
         #ifdef DEBUG
         Serial.print(" Green mode1  ");
         #endif
         redVal = 255;
         greenVal = 0;
         blueVal = 0;
         whiteVal = 0;
         break;
      case 2: //orange
         #ifdef DEBUG
         Serial.print(" Blue mode1  ");
         #endif
         redVal = 255;
         greenVal = 40;
         blueVal = 0;
         whiteVal = 0;
         break;
      case 3: //yellow
         #ifdef DEBUG
         Serial.print(" Purple mode1  ");
         #endif
         redVal = 255;
         greenVal = 255;
         blueVal = 0;
```

```
        whiteVal = 0;
        break;
     case 4: //green
        #ifdef DEBUG
        Serial.print(" Yellow mode1  ");
        #endif
        redVal = 0;
         greenVal = 255;
        blueVal = 0;
        whiteVal = 0;
        break;
      case 5: //blue
        #ifdef DEBUG
        Serial.print(" Orange mode1  ");
        #endif
        redVal = 0;
        greenVal = 0;
        blueVal = 255;
        whiteVal = 0;
        break;
     case 6: //violet
        #ifdef DEBUG
        Serial.print(" White mode1  ");
        #endif
        redVal = 255;
        greenVal = 0;
        blueVal = 255;
        whiteVal = 0;
        break;
      default:
        #ifdef DEBUG
        Serial.print(" Default mode1  ");
        #endif
        redVal = 255;
        greenVal = 255;
        blueVal = 255;
        whiteVal = 255;
  }
  //obtain the potVal
 switch(mode2) {
     case 0:  //100%
        potVal=1;
        break;
     case 1:  //80%
       potVal=0.8;
       break;
     case 2:  //60%
        potVal=0.6;
         break;
     case 3:  //40%
        potVal=0.4;
        break;
     case 4:  //20%
        potVal=0.2;
        break;
    default:
        potVal = 1;
  }
  //next adjust the values to be the correct overall intensity
  redVal = redVal*potVal;
  greenVal = greenVal*potVal;
  blueVal = blueVal*potVal;
  whiteVal = whiteVal*potVal;
}
////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                        Set LED PWM Signals  (DONE)                     //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
void setLEDs()
{
```

```
  analogWrite(ledR,redVal);
  analogWrite(ledB,blueVal);
  analogWrite(ledW,whiteVal);
  analogWrite(ledG,greenVal);
}
////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                        Read Temperature Sensors  DONE                  //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
void tempInput()
{
  //read the Heat Source Temperature Sensor and put it as next vector index
  float heatRead = analogRead(tempHeat);
  float heatVolt = heatRead / 1023.0 * 5000.0;
  heatTemp[heatIndex] = ((heatVolt-750.0) / 10.0)+25.0;
  heatIndex = (heatIndex + 1) % 8;
  heatTempAvg = avgHeatTemps();
  //read the Lava Lamp Temperature Sensor and put it as next vector index
  float lampRead = analogRead(tempLamp);
  float lampVolt = lampRead / 1023.0 * 5000.0;
  lampTemp[lampIndex] = ((lampVolt-750.0) / 10.0)+25.0;
  lampIndex = (lampIndex + 1) % 8;
  lampTempAvg = avgLampTemps();

  #ifdef DEBUG
  Serial.print(" heatTemp: ");
  Serial.print(heatTempAvg);
  Serial.print(" lampTemp: ");
  Serial.print(lampTempAvg);
  Serial.print("  ");
  //Serial.print(" heat Voltage: ");
  //Serial.print(voltageH);
  //Serial.print(" lamp Voltage: ");
  //Serial.print(voltageL);
  //Serial.print("\n");
  #endif
}
////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                        Function to calculate avg heat temp DONE        //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
float avgHeatTemps()
{
  float output = 0;
  for(int i=0; i<8; i++)
  {
    output = output + heatTemp[i];
  }
  output = output / 8.0;
  return output;
}
////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                        Function to calculate avg lamp temp DONE        //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
float avgLampTemps()
{
  float output = 0;
  for(int i=0; i<8; i++)
  {
    output = output + lampTemp[i];
  }
  output = output / 8.0;
  return output;
}
////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                       Calculate Heat PWM Signals    DONE               //
```

```
//                                                                          //
//////////////////////////////////////////////////////////////////////////////
void calcHeat()
{
  //check safety condition on heat source
  if(heatTempAvg > maxHeatTemp)
  {
    heatVal = 0;
  }

  //ON/OFF Controller
  if(50.0 <= lampTempAvg)
  {
    heatVal = 0;
  }
   if(1000.0 >= lampTempAvg)
  {
    heatVal = 255;
  }
}
//////////////////////////////////////////////////////////////////////////////
//                                                                          //
//                          Set Heat PWM Signals                            //
//                                                                          //
//////////////////////////////////////////////////////////////////////////////
void setHeat()
{
  analogWrite(heat,heatVal);
}
```