ADAPTIVE AUTOMOTIVE LIGHTING SYSTEMS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Marc Haeberlin

February 2015

COMMITTEE MEMBERSHIP

TITLE: Adaptive Automotive Lighting Systems

AUTHOR: Marc Haeberlin

DATE SUBMITTED: February 2015

COMMITTEE CHAIR: Dr. Taufik Ph.D.

                                    Professor of Electrical
                                    Engineering

COMMITTEE MEMBER: Dr. Vladimir Prodanov Ph.D.

                                    Assistant Professor of Electrical
                                    Engineering

COMMITTEE MEMBER: Dr. Ahmad Nafisi Ph.D.

                                    Professor of Electrical Engineering

ABSTRACT

Adaptive Automotive Lighting Systems

Marc Haeberlin

Over the past few years, the use of LEDs within the automotive and avionic industries

has increased due to their high efficiency, durability and wide range of light brightness.

As the use of LEDs within these industries grows, a need for reliable, high performance

drivers becomes more relevant. Companies are implementing LEDs for applications

involving adaptive lighting or simple dimming features. This thesis shows

implementation of various non-isolated analog converters integrated with digital dimmers

to achieve these adaptive lighting systems. Adaptive lighting systems involve reading an

input from an external source (brake pedal or steering wheel) and changing the brightness

and/or pattern of the brake/headlights to convey more information to the driver and their

surroundings. The analog converters will implement Linear Technology's LED driver

IC's, while the digital dimmers comprise of microcontrollers and discrete components.

The design, simulation, and hardware verification will showcase the abilities of these

analog converters. Results will demonstrate the proposed applications for both adaptive

front and brake lighting.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 : Introduction

When dealing with lighting applications, there are various types of lights to choose from: incandescent, fluorescent, or LED lights. Traditionally, incandescent lights have been used for applications involving illumination. This type of light bulb emits energy in the form of light from the thermal reaction between a filament and the electric current that passes through it. Incandescent bulbs can be implemented with AC or DC power and no external regulating equipment. Also, low price makes these bulbs desirable, which is why its use in residential and industrial applications has grown in the past. However, the heat lost during the production of light makes incandescent lights energy inefficient. Another option is fluorescent light bulbs. Instead of using a thermal reaction to produce light, like incandescents, fluorescent bulbs utilize a phenomenon known as fluorescence. By passing an electric current through a gas, UV light is emitted that makes the coating of the bulb glow. Fluorescent bulbs are more efficient than incandescent, typically using 15W of power rather than 60W of a standard incandescent to create the same amount of light. In more recent years, LED technology has created more efficient, compact devices, making them more desirable in the field of illumination. LEDs, or light emitting diodes, use semiconductor materials that exhibit electroluminescence, the phenomenon of a material emitting light when an electric current is passed through it [1]. The radiative recombination of electrons and holes creates a light in which the color is dependent on the band gap of the specific material. Today, a wide range of materials is used to create a variety of colors for various applications. LED lights are much more efficient than their counterparts, only taking 6-9W to create the same amount of light output. Although LEDs may have a more expensive initial cost, the lifespan, robustness,

lower maintenance costs, smaller size and efficiency of these lights are making them a more desirable option for illumination and other applications.

A Russian scientist named Oleg Losev first invented LEDs in 1927 [5]. He discovered that the light from the electroluminescence was only low intensity infrared light. Initially, LED had no real useful applications. It was only in 1957 when LEDs became useful for applications involving communication through infrared light at short distances. It wasn't until 1962 when Nick Holonyak, Jr. at General Electric produced the first visible spectrum (red) LED. However production of this new technology was very expensive, totaling $200 per LED produced, making the practical use of LEDs very limited. In 1968, Monsanto Corporation was the first company to mass-produce visible light LEDs, making them usable in electronic applications. GaArP (Gallium Arsenide Phosphorus) was the material used by Monsanto to create red LEDs [2]. As technology for LEDs got more advanced, light output grew, making them better alternatives for illumination applications than fluorescent and incandescent bulbs. Advancements in LED light output follow Haitz Law, which is similar to Moore's law. It states "that every decade, the cost per lumen (unit of useful light emitted) falls by a factor of 10, and the amount of light generated per LED package increases by a factor of 20, for a given wavelength (color) of light." Haitz Law can be shown through a graphical representation in figure 1-1 below.
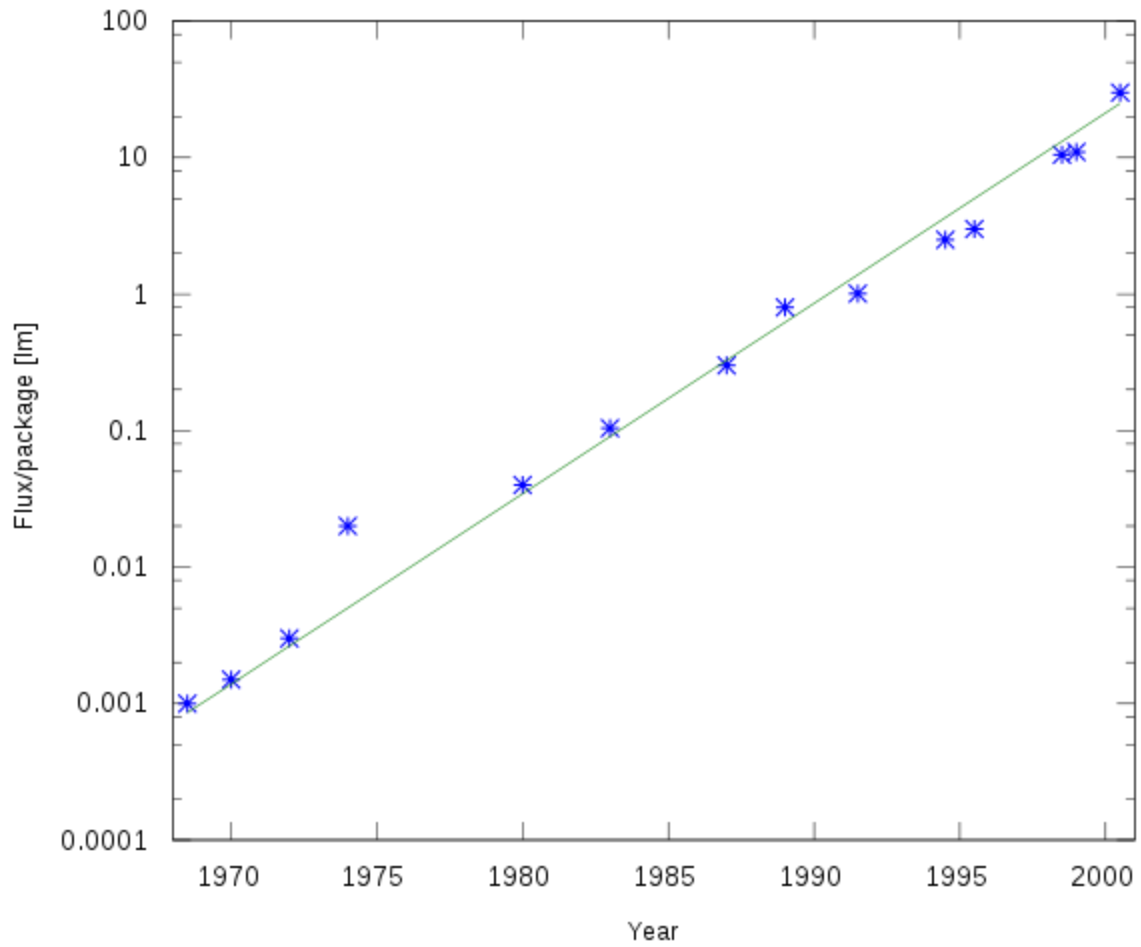
FIGURE 1-1: LED LIGHT OUTPUT PROJECTION PER YEAR [2]

Today, LEDs are used in applications involving: illumination, sensors, advanced communication technology, and visual signals.

Chapter 2 : Background

2.1 Physics of LEDs:

      LEDs are light emitting diodes that are made up of a p-n junction. A p-n junction

consists of a semiconductor material doped with impurities. Current flows from the p-

side to the n-side of the material, but not in the reverse directions [2]. As mentioned

before, LEDs emit light through a phenomenon called electroluminescence. This is the

emission of light through the p-n junction when an electric current passes through it.

When an electron encounters a hole, it falls into a lower energy level and releases energy

in the form of a photon, or light. The wavelength of the energy that is emitted is

dependent on the band gap of the material used in the p-n junction. Figure 2-1 shows an

illustrative example of current flow and light emission through an LED.
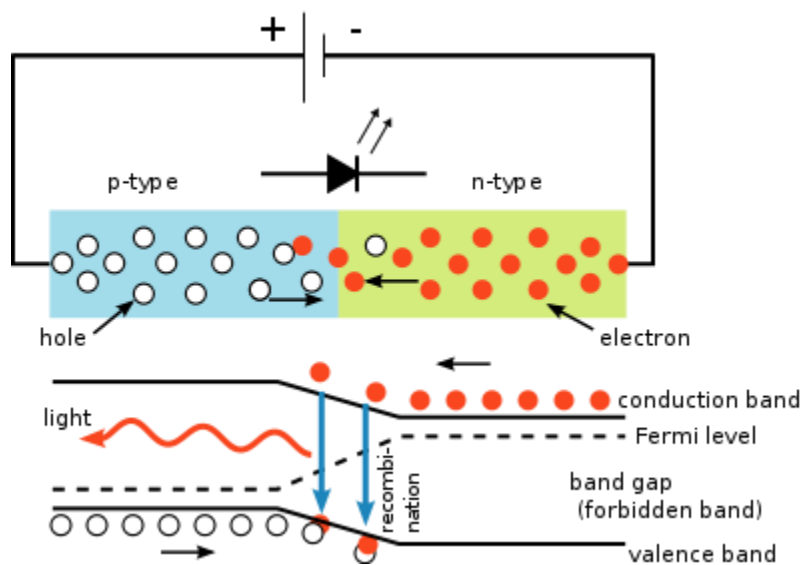
FIGURE 2-1: ELECTRON PHYSIC OF AN LED [2]

There is a wide range of materials used in today's production of LEDs to create various colors. Initially, gallium arsenide (GaAr) was used to create infrared diodes due to its high electron mobility and low device parasitics. As technology advanced, materials that emit shorter wavelengths were developed to cover a wider range of the light spectrum. Figure 2-2 shows the wavelengths and typical materials used to create different colors of light.

Color	Wavelength [nm]	Voltage drop [ΔV]	Semiconductor material
Infrared	$\lambda > 760$	$\Delta V < 1.63$	Gallium arsenide (GaAs) Aluminium gallium arsenide (AlGaAs)
Red	$610 < \lambda < 760$	$1.63 < \Delta V < 2.03$	Aluminium gallium arsenide (AlGaAs) Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
Orange	$590 < \lambda < 610$	$2.03 < \Delta V < 2.10$	Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
Yellow	$570 < \lambda < 590$	$2.10 < \Delta V < 2.18$	Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
Green	$500 < \lambda < 570$	$1.9^{[70]} < \Delta V < 4.0$	**Traditional green:** Gallium(III) phosphide (GaP) Aluminium gallium indium phosphide (AlGaInP) Aluminium gallium phosphide (AlGaP) **Pure green:** Indium gallium nitride (InGaN) / Gallium(III) nitride (GaN)
Blue	$450 < \lambda < 500$	$2.48 < \Delta V < 3.7$	Zinc selenide (ZnSe) Indium gallium nitride (InGaN) Silicon carbide (SiC) as substrate Silicon (Si) as substrate—under development
Violet	$400 < \lambda < 450$	$2.76 < \Delta V < 4.0$	Indium gallium nitride (InGaN)
Purple	Multiple types	$2.48 < \Delta V < 3.7$	Dual blue/red LEDs, blue with red phosphor, or white with purple plastic
Ultraviolet	$\lambda < 400$	$3.1 < \Delta V < 4.4$	Diamond (235 nm)[71] Boron nitride (215 nm)[72][73] Aluminium nitride (AlN) (210 nm)[74] Aluminium gallium nitride (AlGaN) Aluminium gallium indium nitride (AlGaInN)—down to 210 nm[75]
Pink	Multiple types	$\Delta V \sim 3.3^{[76]}$	Blue with one or two phosphor layers: yellow with red, orange or pink phosphor added afterwards, or white phosphors with pink pigment or dye over top.[77]
White	Broad spectrum	$\Delta V = 3.5$	Blue/UV diode with yellow phosphor

FIGURE 2-2: WAVELENGTH AND SEMICONDUCTOR MATERIAL USED FOR LEDS [2]

2.2 Implementation:

LEDs follow an exponential relationship between current and voltage. A small change in forward voltage may result in a large change in forward current. Typically, the forward voltage of an LED ranged from 1.9-4.2V, depending on the color. Figure 2-3 [16] shows the I-V characteristics of an LED.
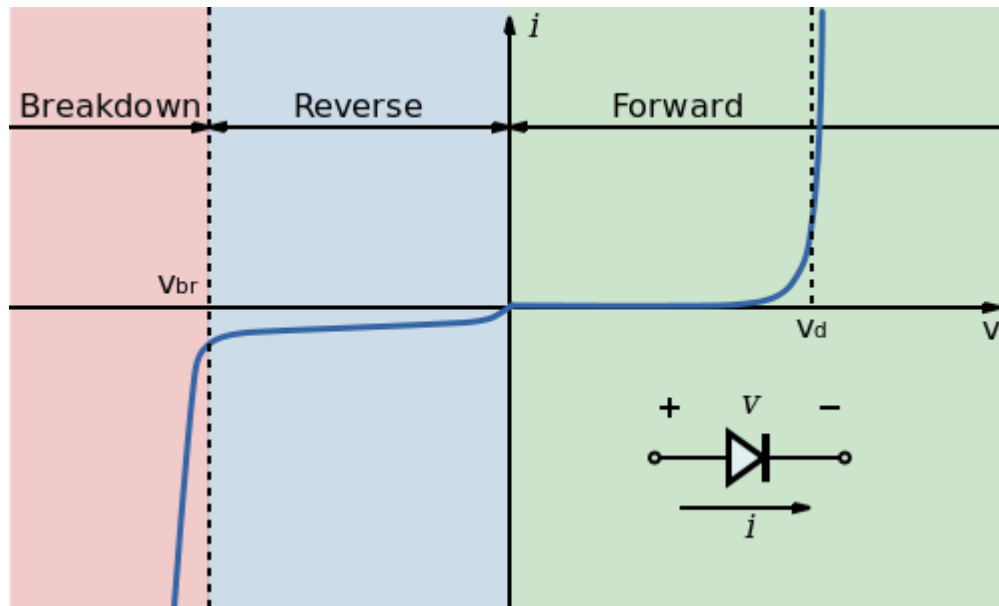
FIGURE 2-3: I-V CHARACTERISTICS OF AN LED [16]

Due to this exponential relationship, driving an LED becomes more complicated than a fluorescent or incandescent light bulb. Properly biasing an LED requires a stable and reliable driver that acts as a constant current source. For simple applications, this can be done with a single resistor circuit, but typically for standard applications that turn on/off or alter the amount of forward current, an LED driver circuit that monitors the output current is needed to maintain proper operation of the circuit. Power electronics circuits, or switching-regulators are used to either step up or step down the source voltage depending on the forward voltage of the LED string.

Development in both areas of LED and its drivers has allowed LEDs to be used in a wide range of applications. Visual indicators, illumination and data communication are some examples. This paper will discuss the use of LEDs as visual indicators in automotive applications implemented using non-isolated DC/DC converters and microcontrollers.

2.3 DC/DC Converters - LED Drivers:

To properly drive an LED, precise current control is required. There are a few things to consider when choosing a driver. Obviously, input and output current and voltage requirements place constraints on the system. Depending on the application, the load may have multiple LEDs configured in series, changing the overall output voltage of the load. A more complicated application involving dimming the LEDs requires a fast and stable control loop to ensure the forward current through the LEDs remains at a safe level. In the past, linear regulators were a popular option, but due to their low efficiency, industry has switched to switching-mode regulators. DC/DC converters acting as constant current source have become a desirable option to those looking for visual versatility and efficiency within their LED applications. There are a wide variety of different DC/DC converter topologies. The two main groups are categorized as non-isolated and isolated; this thesis will be focusing primarily on non-isolated topologies such as buck (step-down), boost (step-up) and combinations of the two.

In a typical DC/DC converter, output voltage is monitored. An LED driver requires the output current to be monitored, as the forward voltage is a function of the forward current. Because of this, the load does not need to be connected to ground and can act properly in a floating configuration.

Figure 2-4 shows the standard buck circuit compared to a floating load buck converter. Although the topology is slightly different, the overall function and performance remains the same.
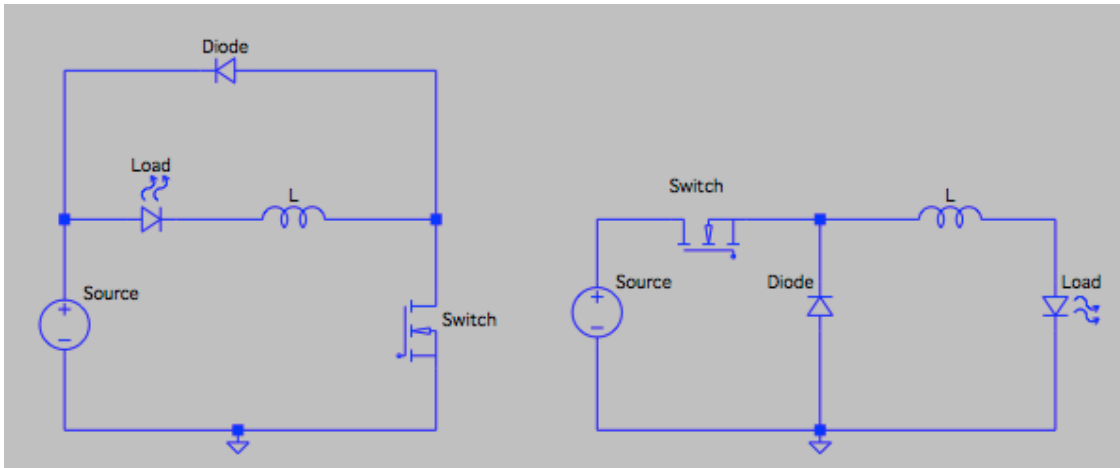
FIGURE 2-4: BUCK CIRCUIT WITH FLOATING LOAD(LEFT) AND GROUNDED LOAD(RIGHT)

To monitor the output current, a small valued sense resistor is placed in series between the load and the converter. An IC monitors the voltage across this sense resistor and will adjust the duty cycle of the converter if the voltage across that resistor changes. However, the control loop must take into account the dynamic resistance of the specific LED of the application. Because the I-V characteristics are nonlinear and change depending on the average forward current, an operating point must be set for the converter to properly maintain output current [4]. A simple diagram of the control loop is shown below, in figure 2-5.

Figure 4. Small-signal model of current-regulating boost converter

(a) Complete

(b) Simplified
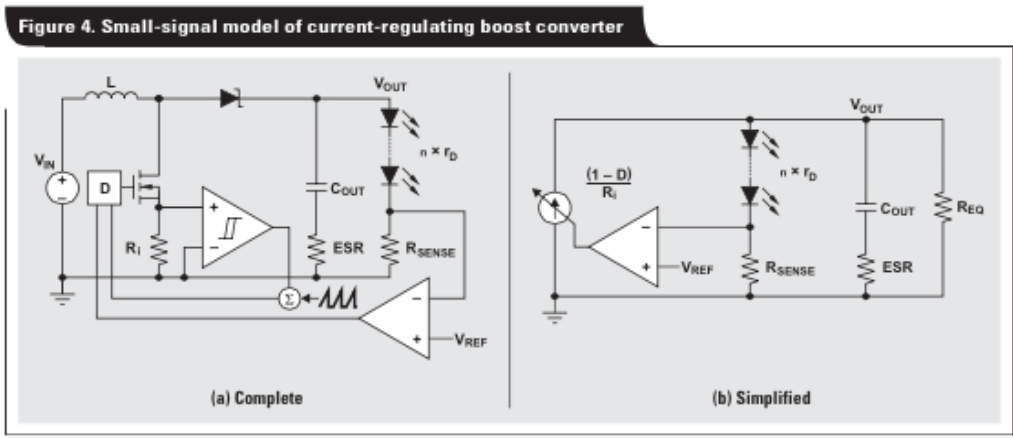
FIGURE 2-5: CONTROL LOOP OF DC-DC CONVERTER [4]

Although LEDs follow the I-V characteristics seen in figure 2-3, it is unwise to configure LEDs in parallel. The forward voltage of one LED in the parallel configuration may change based on the differing device or thermal characteristics, making it susceptible to thermal runaway [2]. Also, short and open faults yield this configuration to high surges of currents through a single LED that may destroy it. To avoid this, LEDs are always configured in series, also known as a "string" of LEDs.

For application involving a set forward current, monitoring the output current becomes relatively easy. Resistor biasing or DC/DC converters with low bandwidth are easy solutions for simple applications. However, for application that involve changing the intensity of light output, or dimming the LEDs, the converter becomes more complicated. Before I go any further, I want to talk about possible ways to dim the LEDs. In industry, there are two main ways to adjust the brightness: analog dimming and pulse-width modulation (PWM) dimming [3].

9

Analog dimming is a simpler method that involves changing the average forward current to adjust intensity. Adjusting the value of the sense resistor using a small value potentiometer can achieve this. However, performance characteristic of this method make it an undesirable option for more complicated applications. For example, when running identical LEDs with the same forward current, the color may not be the same. Also, analog dimming cannot achieve the same range of light brightness compared to PWM dimming.

PWM dimming involves transitioning the LED to 100% intensity to full off at a fixed frequency with variable on time. There are two ways to do this. Two-wire PWM dimming involved modulating V_{IN} so that V_{OUT} modulated accordingly, shutting off the current through the LEDs. This method is not very desirable because the input source must be able to provide a PWM signal [13]. The method I will be focusing on in this thesis is PWM dimming with a shunt device. This is achieved by placing a series or parallel shunt switch across the LEDs to decrease the average current through the LED [13]. A series shunt turns off the output current of the converter, in turn shutting off the current through the LEDs. A parallel shunt bypasses the LED current through the switch. However, one must take into account the power loss from the on resistance of the switch when dealing with a parallel shunt. This switch is typically a PMOS driven by a high-side driver that is controlled by an external PWM signal. Because LEDs have the ability of very fast turn on and off time, the on/off switching is unnoticeable so long as the PWM switching frequency is above 100 Hz. This means the LED is turning on and off 100 times per second. This frequency is the limit to which the human eye can noticed this switching. Although for less intense ambient light (at night), the eye becomes more

sensitive to the switching and can see the flickering of the light up to 120Hz. This method of dimming allows for very high PWM dimming ratios. A PWM dimming ratio is typically expressed in the following way: ((Period/On-Time): 1). There are technology that have achieved upwards of 20,000:1 dimming ratios for avionic applications that require very dim interior light due to the low intensity of the ambient light in the cockpit. Although PWM dimming allows for a wider range of light intensity, the need for external logic for the PWM signal increases the complexity of the system.

It is possible to have a string of LEDs where individual LEDs can be controlled by a unique PWM signal. By placing parallel shunt switches across each LED, along with some discrete logic, a single driver can be used to control multiple LEDs. However, there are some issues that arise when dealing with this configuration. For example, when running a string of 8 LEDs, there is a set output voltage seen by the converter determined by the on-voltage of the LED string. If one or more of the parallel shunt devices were to turn on, this changes the output voltage. The most severe case would be the transition from 8 LEDs on to all off. This is a huge change in output voltage. With a slow control loop, this could result in the converter thinking that no current is flowing into the LEDs (due to the LED I-V characteristics discussed earlier), resulting in a high surge of current pumped to its output. Likewise, when switching from no LEDs on to all on, a dip in output current will occur, resulting in a short off time in the brightness of the LEDs. Now, this wouldn't be too big of an issue with a circuit that has an optimized control loop, which I will discuss later. However, another issue arises when configuring LEDs in a string. There is typically an output capacitor placed across the LED string that will slow the change in voltage at the output. This capacitor limits the reaction of the circuits

control loop and the circuits ability for high PWM ratio. To avoid this, the output capacitor may simply be removed. By doing so, the output voltage is not "held" at its current value and can change much more rapidly, shortening the response time of the circuits control loop. Although an output capacitor may be useful as an AC current filter and can reduce the size of the inductor, the pros outweigh the cons [12]. A perfect current source has infinite output impedance, which allows the output voltage to change very rapidly to changes in output current. By removing the output capacitor, the output impedance is a function of the inductor and input and output voltages [12].

Now, the output capacitor is not removed from the circuit altogether. As mentioned before, there is a sense resistor to monitor the output current. By placing a small valued capacitor (that may be adjusted for control loop optimization) across this sense resistor, the voltage seen by the converter is now filtered to a DC signal. This allows the circuit to function properly. Even though the output current through the LED string is now unfiltered, it is equal to the inductor current and contains high ripple. The switching frequency of the converter is much higher than the 120Hz limit, so the ripple is not seen by the naked eye.

To maintain proper output current in a converter, an optimized control loop is necessary. It is the control loops job to monitor the LED current, through the sense resistor discussed earlier, and to adjust the duty cycle of the switch so that big fluctuations in output current or voltage do not occur. The faster the control loop, the faster the circuit is able to react in changes in output voltage, resulting in high PWM dimming ratios [4]. The sense resistor is the first step in this control loop. A comparator monitors this voltage against its reference voltage and the output signal is sent to another

comparator. This second comparator has a triangular wave reference signal. Depending on the value of the output of the first comparator, the output of the second comparator will adjust accordingly to control the duty cycle of the converter. This will change the amount of output current. This method of adjusting the duty cycle of the main switch is called slope-compensation.

In a typical LED driver, a resistor and capacitor connected to a pin on the IC are used to vary the control loop speed, making the IC more versatile for various applications. This pin is usually labeled as V_C, for compensation. The main goal of this RC circuit is to cause an increase in voltage at this pin. The RC circuit determines this speed at which this voltage change occurs. A larger capacitor will draw less current through the branch. A larger resistor will cause a faster voltage change seen by the pin. So ideally, for a very fast control loop, one would set the capacitor to a very small value (even 0 Farads) and the resistor to a very large value. Alternatively, for a very slow control loop, a very small resistor (0 ohms) and a big capacitor would be placed at this pin.

Now that the basics of LED drivers have been covered, it is important to discuss how the signal controlling the PWM switch will be generated. Because PWM waveforms are square waves with set frequency and variable duty cycle, a microcontroller can be used to generate these signals from its GPIO pins. Depending on the number of shunt switches, a particular microcontroller may be chosen based on the number of GPIO pins. Also, the resolution of the ADC will be an important factor, which is discussed later on.

For this thesis, I will be exploring different ways to achieve adaptive brake and headlight systems for automobiles using LED drivers and microcontrollers. Adaptive

brake lights are brake lights that change in brightness and/or pattern based on the brake

pedal position. Adaptive headlights change the angle of the beam based on steering wheel

position in order to light up more of the road. This is helpful when a car is turning and

will illuminate the sidewalk or other objects that could obstruct the vehicle. Both of these

systems are an attempt to give drivers more information about their surroundings, making

the road a safer place. To simulate an automobiles input, like a steering wheel or brake

pedal, a potentiometer will be used to generate this signal. This will involve designing

LED drivers for various loads, some with multiple LEDs in a string and some with a

single LED per driver, to compare price and performance characteristics of different

designs. By incorporating the knowledge within this chapter, I will design application

circuits around various Linear Technology IC's, optimize the control loop, and create

digital PWM signals using microcontrollers to create a system using both analog and

digital components.

Chapter 3 : Design Requirements

As mentioned in the earlier chapters, the goal of this thesis is to design and build the analog and digital components of the adaptive lighting systems using Linear Technology IC's. To do so, I will be configuring three different topologies using three different IC's that vary in performance and price.

- *Topology 1: High Price and High Performance*

For the first topology, I will attempt to build a high performance, but high cost driver that allows for very high PWM dimming ratios and individual control over each LED.

- *Topology 2: Low Price and Medium Performance*

The next topology will use only 1 LED driver to drive a string of LEDs. This will minimize the parts required, however reducing the speed and performance of the driver. However, a solution to control individual LEDs will need to be created to allow for full control of the load.

- *Topology 3: Low Price and Medium Performance*

The final topology built will also only have 1 LED driver to drive a string of LEDs. A different IC will be used to demonstrate its capabilities.

However, all three of these topologies will be integrated within an automotive system, so it is important to recognize the systems limitations. In an automobile, the power source is limited to the car battery. A fully charged battery will have a nominal voltage of 12.6V. During the charging cycle of the battery, the alternator boosts the voltage up to 13.8V - 14.4V. A fully discharged battery will decrease voltage to 12V. To

ensure proper operation of my driver circuits, I will assume an input voltage range of 9-16V.

The output voltage of the system will be dependent on the size of the load, or the LED string. To simulate a brake/headlight, an array of 12 LEDs will be used. The output voltage can range from 0V - 40V. Because PWM dimming will be implemented, there will be times during the cycle at which no LEDs are conducting, which results in an output voltage of 0V. Similarly, when all LEDs are on, the output voltage consists of the sum of 12 red or white LEDs forward voltages, or 40V.

The amount of output current needed will vary for specific applications. Depending on how bright the LEDs must illuminate for a specific application will change the amount of output current. For brake/headlight applications, an output current of 500mA - 1A will suffice and illuminate the road sufficiently.

In an automobile, an electric system is subject to a wide range of interference in terms of EMI. A car is essentially a large metal cage that picks up signals from the surrounding area. Due to the wide variety of electrical components, such as a radio, it is important to make sure each system is designed to avoid interference with its neighboring systems. Switching regulators emit a large amount of EMI within their switching frequency range. As mentioned earlier, a popular and important system in an automobile is the radio. Radio frequencies can be in the AM or FM band. The AM band ranges from 535-1605 kHz, while the FM band ranges from 88 - 108MHz. To avoid these frequency ranges, the switching frequencies of the analog converters will be set below 500 kHz or above 120MHz. To operate at the higher end of this limit would allow smaller inductor sizing due to the shorter period. However, DC/DC converters have a minimum on time

requirement that vary from converter to converter. When the output of the converter is at 0V (when no LEDs are conducting), this may become an issue. Such a short period may result in improper operation of the converter. For this reason, the switching frequency of the converter will be set on the lower end of the frequency range.

Another system limitation to consider will be the size of the system. This is important to remember because the amount of space allotted to a headlight or brake light in an automobile is very limited. Typically, the inductor is a component that takes up a sustainable amount of space. As mentioned earlier, the switching frequency will be set below 500 kHz, resulting in a larger inductor. To reduce the size of the inductor, the amount of ripple in the inductor current may be increased (see equation 3-1 and figure 3-1). It is standard to allow the ripple current through the inductor to be 5%-30% of the normal operating current. [15] A larger ripple also has an effect on the output capacitor. Because the capacitor provides a low impedance path for high frequency signals, a larger ripple can produce a voltage ripple at the output due to the capacitor's equivalent series resistance. A large output voltage ripple is undesirable for the load if it requires a constant operating voltage. Capacitors with low ESR are more expensive, so this is something to keep in mind.
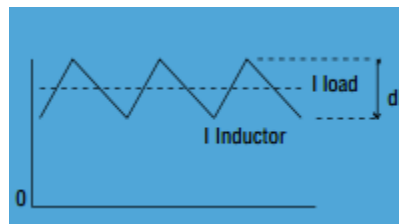
FIGURE 3-1: STANDARD INDUCTOR CURRENT RIPPLE IN CONTINUOUS CONDUCTION MODE

$$\frac{V_L}{L} = \frac{dI_L}{dt} \qquad\qquad\qquad (3\text{-}1)$$

Another main component within a DC/DC converter is the switch. Depending on the operating current and switching frequency will help determine what switch is right for the job. There are two main qualities of a switch to pay close consideration to, the on-resistance and gate charge of the switch. These two components are usually inversely related. A switch with a large on resistance will have a small gate charge, and vice versa. The on-resistance of the switch determines the amount of power lost during conduction, or the on time of your converter. Depending on the duty cycle and output current of the converter, a larger on-resistance may be okay. If the switching frequency of the converter is very high, it is beneficial to have a switch with a small amount of gate charge, allowing it to switch at a faster rate.

The topologies implemented in this thesis will be of the non-synchronous nature. Thus, the catch diode in DC/DC converter is an important component. Since the switching frequency of the converter will be below 500kHz, a "fast" diode is not required. A "fast" diode will have a quick turn-on time, as well as a quick reverse-recovery time. Minimum power loss is ideal, as the diode will dissipate a small amount of power when it is conducting. This conduction loss is determined by the on-resistance of the diode. Another option would be to use a synchronous DC/DC converter. A synchronous converter uses two MOSFETs, instead of one MOSFET and a diode to perform the same capabilities as a standard non-synchronous converter. However, this is a more expensive option as the IC needed to run a synchronous converter is more expensive. Rather than the control loop having to account for a single switching signal, it

would now have to account for two switching signals. This is an unnecessary option for my applications.

To control the LEDs, a square wave with variable duty cycle will be used to control the shunt switch. A microcontroller with GPIO pins will be used. Since this application will involve 12 LEDs, it is important that the microcontroller has at least 12 GPIO pins so that each LED has its own unique PWM signal. Also, to simulate the position of the brake pedal or steering wheel, an external voltage source will be read by the microcontroller. A potentiometer will be used to simulate this external voltage, as the wiper will range from 0V to the supply voltage. An ADC will be required by the microcontroller for this.

In the end, these three topologies will be integrated to work as an adaptive lighting system for automobiles. Performance and price will be examined.

Chapter 4 : Design and Simulation

       In this chapter, the design of the non-isolated DC/DC converters will be configured to run as a constant current source to power a string of LEDs. Three different topologies will be designed (discussed in the previous chapter) using a buck circuit, boost, or a combination of the two to convert the automobiles DC voltage to an output voltage capable of driving the required load. A microcontroller, carrying out commands written in C, will be used to create the PWM waveforms for the corresponding shunt switches to change the brightness of the load. Individual control of each LED will be key when designing this project. After all is said and done, three different topologies with varying performance/price characteristics will all be able to perform similar execution of adaptive light technology. Figure 4-1 shows the black box diagram of the adaptive lighting system that will be designed.
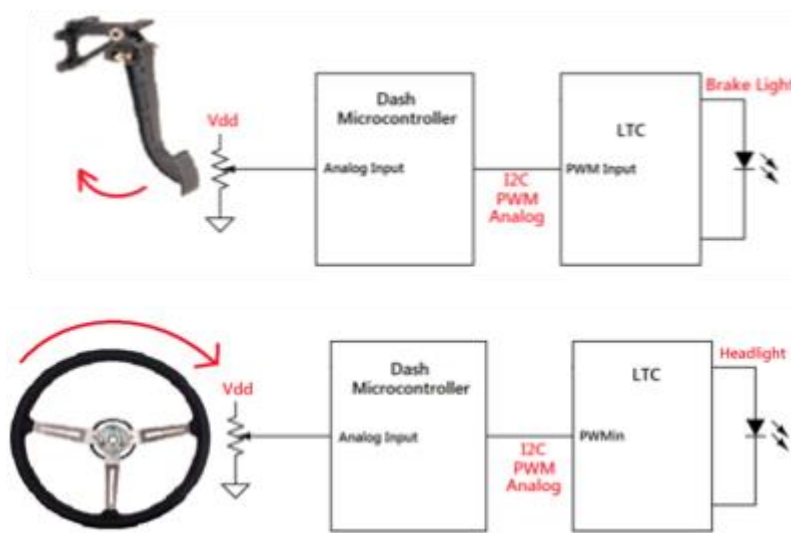
FIGURE 4-1: BLACK BOX MODEL OF ADAPTIVE LIGHTING SYSTEMS

Topology 1: High Price High Performance using LT3797 Buck Mode

For this first topology, each LED will have its own driver. The system

requirements for topology 1 can be seen in table 4-1.

TABLE 4-1: SYSTEM REQUIREMENTS FOR TOPOLOGY 1

V_{IN}	9-16V
V_{OUT}	0-3V
I_{OUT}	1A
$f_{SWITCHING}$	310 kHz
Load	1 Red LED

Because the input will always be greater than the output, a buck topology will be

used. The circuit will drive 1 red LED at 1A. The switching frequency was chosen

(somewhat arbitrarily) based on the constraints presented in chapter 3. As mentioned in

the previous chapter, a standard buck circuit is configured differently than a buck used as

a constant current source. Figure 4-2 shows a constant current buck circuit.

Since the LEDs require a constant current, not a constant voltage, a floating load configuration (not connected to ground) can be used. The main components of a buck are still present: inductor (L), the main switch (Main $_{SW}$) and diode (D). The load (LED) is connected from the input to the inductor. A PWM series shunt switch is connected from the input terminal to the positive node of the LEDs. A series shunt is used in this topology because of the single LED load. This switch is controlled by an external PWM signal that will open and close, allowing pulses of current to pass through the LEDs. The resistor in series with the load, R_LED, controls the DC current through the LEDs. The IC will monitor the voltage across this resistor, so changing the nominal value will change the current through the load. Equation 2 shows the derivation for this resistor. The

feedback network, which comprises of R_FB1, R_FB2 and the output capacitor, C_{LOAD},

ensures for over voltage protection. OVP (over voltage protection) is necessary in the

event an LED breaks to an open. Not only does it protect the LEDs from exceeding the

maximum forward voltage, but also the PWM switch. However, for a buck, OVP is not

as important because the maximum output voltage is the input voltage. This is not the

case for a boost circuit. Lastly, the resistor in series with the main switch, R$_{LIM}$, sets the

maximum switch peak current allowed by the circuit. The current through the switch will

have a ripple that is set by the inductor, so R$_{LIM}$ ensures a safe level of current through

the switch during the charging cycle.

Although this looks slightly different from the standard topology, the charging

and discharging cycles look similar. Figure 4-3 shows the charging cycle of the buck.
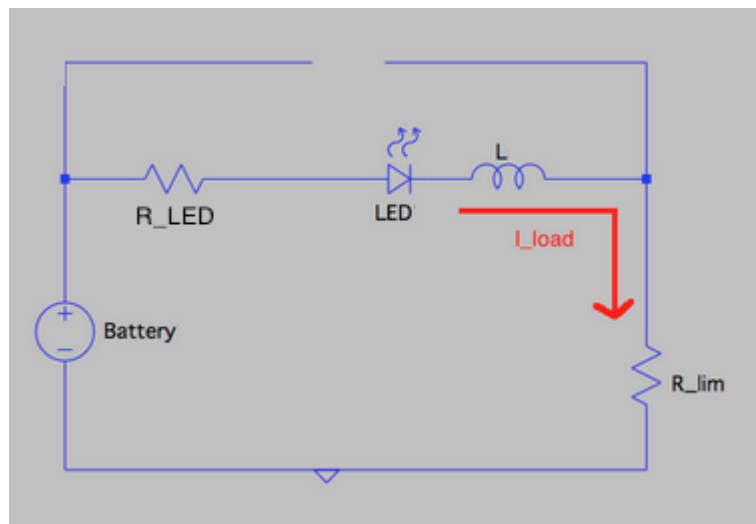
FIGURE 4-3: CHARGING CYCLE OF BUCK CONVERTER

When the main switch in conducting, the diode is reversed biased and does not

conduct current. Current flows from the input, through the load, inductor and main

23

switch, just like a standard buck. The voltage across the inductor is positive (seen in

equation 4-1), charging the inductor.

$$V_{L\ Charging} = V_{IN} - V_{OUT}$$

$$where\ V_{OUT} = V_{LED}$$

During the discharging cycle of the converter, the switch is off (open circuit) and

the diode is forward biased. Figure 4-4 shows the discharge cycle of the buck converter

and equation 4-2 shows the voltage across the inductor during this time.
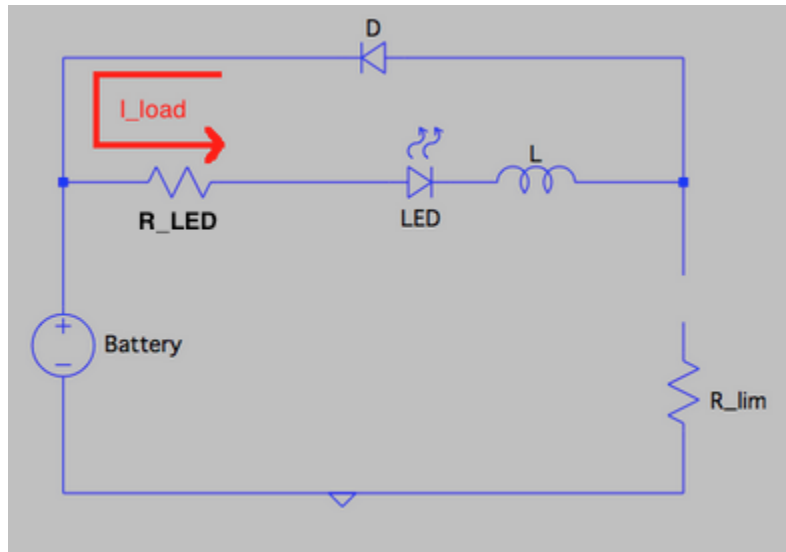
FIGURE 4-4: DISCHARGING CYCLE OF BUCK CONVERTER

$$V_{L\ Discharging} = V_{IN} - V_{OUT} - V_{IN} = -V_{OUT}$$

$$where\ V_{OUT} = V_{LED}$$

Positive voltage across the inductor causes the current through the inductor and

MOSFET to linearly increase, or charge up. When the switch turns off, the voltage across

the inductor is now negative and discharges through the diode, still circulating through

the load. The inductor and switch current exhibit the same triangular waveform as a typical buck, shown in figure 4-5.
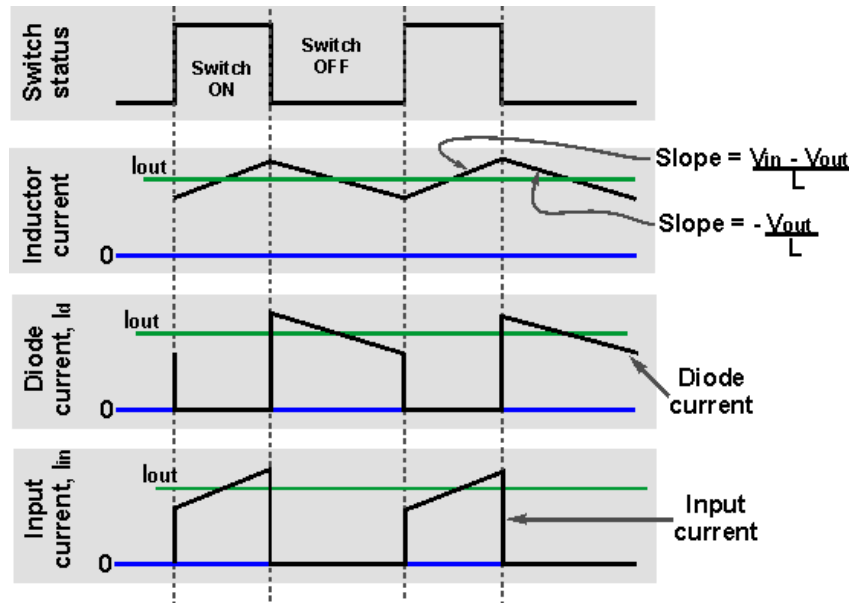
FIGURE 4-5: BUCK CURRENT WAVEFORMS FOR CHARGING AND DISCHARGING CYCLE

Current through an inductor follows the relationship shown in equation 3-1. By allowing 700mA peak-to-peak ripple, or around 35% of the average output current, through the inductor, a 12uH inductor can be used. Equation 4-3 shows the derivation of this calculation.

$$\frac{V_L}{L} = \frac{dI_L}{dt} \Rightarrow L = (V_{IN} - V_{OUT}) * \frac{\Delta t_{ON}}{\Delta I_L} = \frac{(16V - 3V) * 0.6us}{700mA}$$

$$where \; \Delta t_{ON} = DT = \frac{V_{OUT}}{V_{IN}} * T = \frac{3}{16} * 3.2us \qquad (4\text{-}3)$$

$$L \approx 12uH$$

However, this inductance value does not take into account the current ratings for the device. Inductors have two different current rating; the saturation current and the

RMS current. The saturation current indicates the limit at which the device will keep its nominal inductance value. Past this value, the inductance of the device decreases. When choosing this value, it is important to choose a value that is sufficiently higher than the average output current. The RMS value takes into account the heat dissipation the device can withstand. Typically, these two values are relatively close to each other. Since the average output current of this topology will be at 1A, I_{SAT} and I_{RMS} should be around 1.5-2A.

Along with the inductor, both switches must be sized. To do so, the maximum voltage across each switch must be calculated. A poorly sized switch will most likely break and cause problems for your circuit. The main switch, or Main_SW in figure 4-2, will have a maximum voltage when it is not conducting. This occurs during the discharging cycle. Equation 4-4 shows the derivation.

$$V_{MAIN\ SWITCH\ (MAX)} = V_{IN} = 16V$$

(4-4)

The PWM switch follows the same process. The maximum voltage across the switch will be when it is off and not conducting. This can occur during the charging or discharging cycle, so it is important to look at both cases for the maximum switch voltage. Equation 4-5 shows the derivation.

$$V_{PWM\ SWITCH\ (MAX)} = V_{OUT} = V_{LED}$$

(4-5)

Next is sizing the diode. Again, a poorly sized diode can mean death for a circuit. Similarly to the switches, the maximum forward voltage occurs when the diode is reverse biased and not conducting. This occurs during the charging cycle. Equation 4-6 shows the derivation.

$$V_{DIODE\ (MAX)} = V_{IN} = 16V$$

<div align="right">(4-6)</div>

Now that the main components are sized, I will pick an IC to work with. The IC I will use for this topology will be the LT3797. This is a triple-channel LED driver that satisfies my design requirements. Input voltage can range from 2.5V - 60V with an output voltage up to 60V. This IC can run a load up to 35W, which is more than sufficient for this application. Triple-channel LED driver means that on a single IC, there are essentially three drivers, allowing 3 separate loads that can be individually controlled. To run 12 LEDs, 4 LT3797's will be used each running a 3 LEDs per IC. This IC can sense the output current at the high side of the load, ensuring a quick shut off if necessary. It also has OVP. A single channel of the LT3797 circuit is shown in figure 4-6. As you can tell, the LT3797 is configured as a constant current source, exactly like figure 4-2.

To obtain the correct level of output current, the resistor between the ISP and ISN nodes must be sized according to equation 4-7 [9].

$$I_{OUT} = \frac{250mV}{R_{LED}} = 1A \Rightarrow R_{LED} = 250m\Omega$$

<div align="right">(4-7)</div>

To ensure the peak current through the switch does not cause damage to the switch, R$_{SWITCH}$, resistor between Sense$_P$ and Sense$_N$, must be sized according to equation 4-9. An allowable peak current must be greater than the nominal peak. Equation 4-8 shows the derivation.

$$I_{OUT\ AVERAGE} + \frac{I_{PEAK}}{2} < I_{MAX\ PEAK} > 1A + 350mA = 1.35A$$

<div align="right">(4-8)</div>

The peak output current will be set to 1.6A. Equation 4-9 [9] calculates the value of the switch resistor with this peak output current.

$$R_{SWITCH} = \frac{80mV}{I_{SWITCH\ PEAK}} = \frac{80mV}{1.6A} = 50m\Omega \tag{4-9}$$

Next is sizing the feedback resistors. For the LT3797, the IC monitors the voltage from ISP to FB, or the voltage across R1. Once this voltage reaches 1.25V, the IC shuts off current to the load. Unlike normal voltage regulated loads, the feedback network ensures that the output voltage does not run away in the case of an open LED event. This feedback network is present as a backup to the current limiting resistor, R_{LED}. The feedback network is sized according to equation 4-10 [9].

$$V_{OUT} = 1.25\frac{R_{FB1}}{R_{FB1} + R_{FB2}} \tag{4-10}$$

By setting the OVP to 5V, 2V above the standard forward voltage of a single LED, R_{FB1} and R_{FB2} can be sized as follows:

$$R_{FB1} = 220k, R_{FB2} = 680k$$

The output capacitor placed across the LED and feedback network will determine the speed at which the control loop reacts to the constant change between ON/OFF of the output current during the PWM dimming cycles. This sizing of the capacitor will be discussed later in this chapter.

Figure 4-6 shows a single channel of the LT3797 configured in buck-mode with the calculated component values.
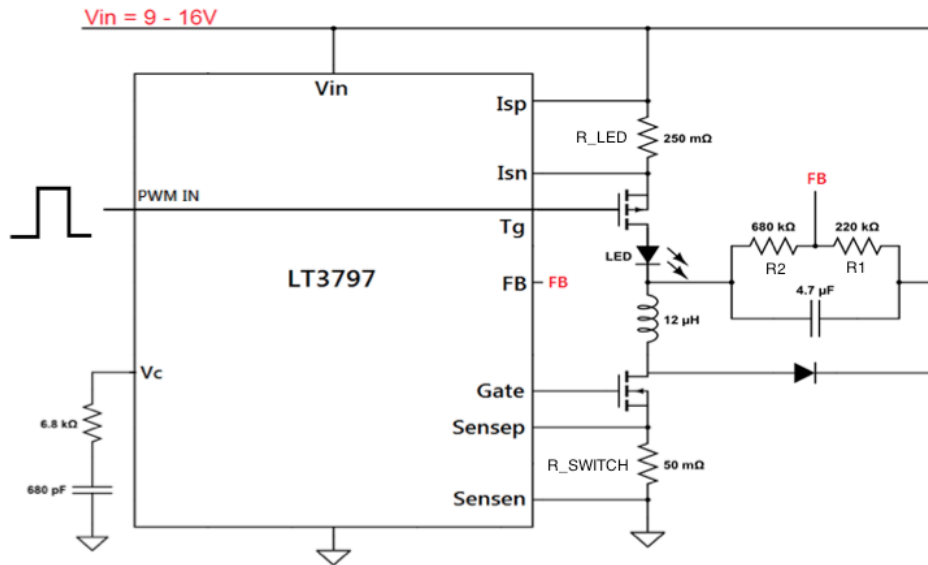
FIGURE 4-6: LT3797 RUNNING SINGLE LED @ 1A

single channel shown for simplicity

Before testing the compensation network, it is important to make sure the circuit

is operating with proper specifications. Figure 4-7 shows the full LT3797 LTSpice

schematic with 3 channels configured in Buck-Mode with an output current of 1A.
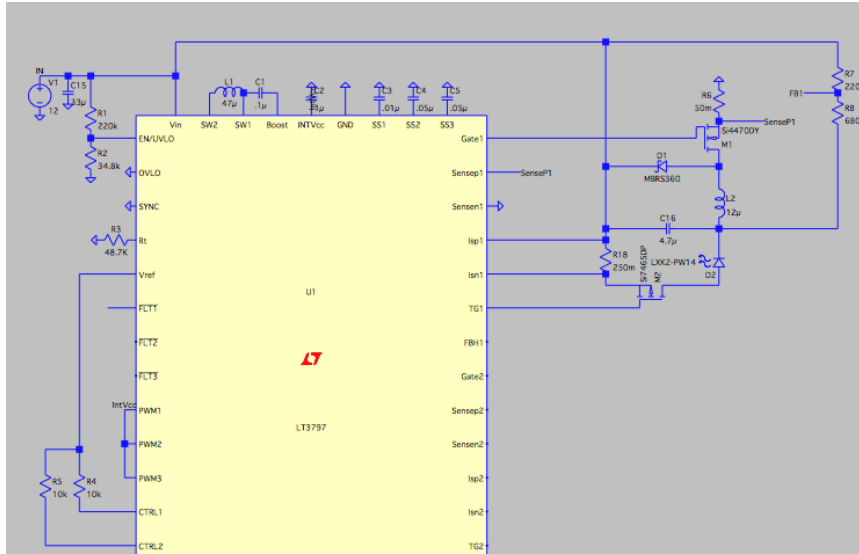
FIGURE 4-7: SINGLE CHANNEL OF LT3797 CONFIGURED AS BUCK

(Note that only one channel was simulated to avoid redundancy)

Figure 4-8 shows the correct output current through each of the LED branches.
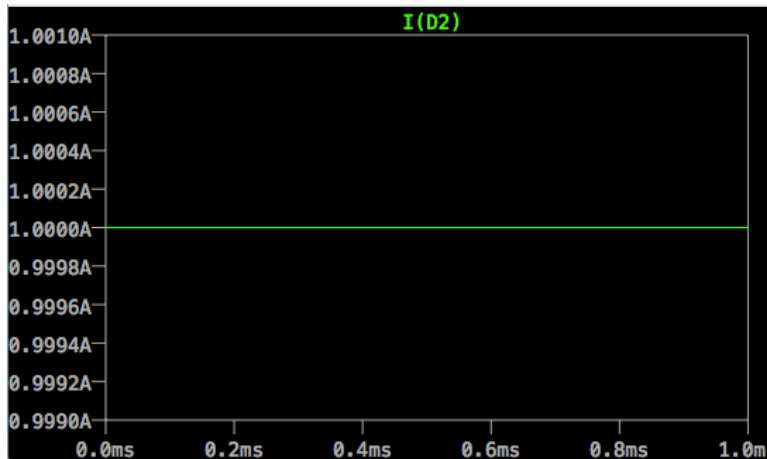
FIGURE 4-8: OUTPUT/LED CURRENT OF 1A FOR LT3797 BUCK DRIVER

Another major component of this design is the compensation network. As mentioned in chapter 3, there is an RC network connected to the V_C pin of the IC. When the values of this RC network are changed, it changes the speed at which the control loop

reacts. This, in combination with the output capacitor, plays a huge roll in the speed of the circuit. Again, speed is very important in this design because it will dictate the PWM dimming ratio that can be achieved. PWM dimming ratio indicates how dim an LED will be able to illuminate while the circuit has sufficient stability. To test the speed of the circuit, I can utilize the CTRL pin on the IC. This CTRL pin allows for analog dimming. The control pin can be biased from 0-1.25V to change the level of output current. For example, when $V_{CTRL} = 0.75V$, the output current will be 50% of the average value. By inputting a square wave that oscillates between 0.75V and 1.25V, the output current of the device will be constantly changing from 50% to 100%. By analyzing the transient response, I can adjust the values of the RC compensation branch as well as the output capacitor to obtain values that optimize the speed of the circuit. This is an easy quick way to test for speed that only looks at the transient response. However, this method does not give exact numerical values, like a bode plot analysis would. Figure 4-9 shows the setup for this test.
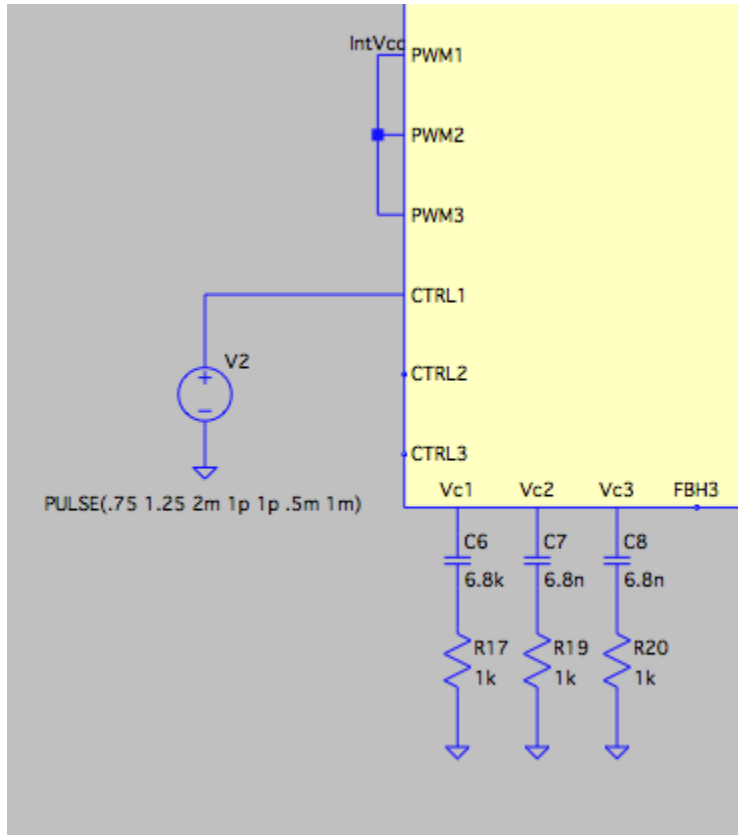
FIGURE 4-9: TRANSIENT RESPONSE TO ANALOG DIMMING SETUP UING LT3797

Figure 4-10 shows the best transient response of the test. The RC network and output capacitors that yielded this transient response can be seen in table 4-2.

TABLE 4-2: CTRL TEST COMPONENT VALUES FOR TOPOLOGY 1

Component	Value
R	5kΩ
C	4.7nF
C_{OUT}	4.7uF
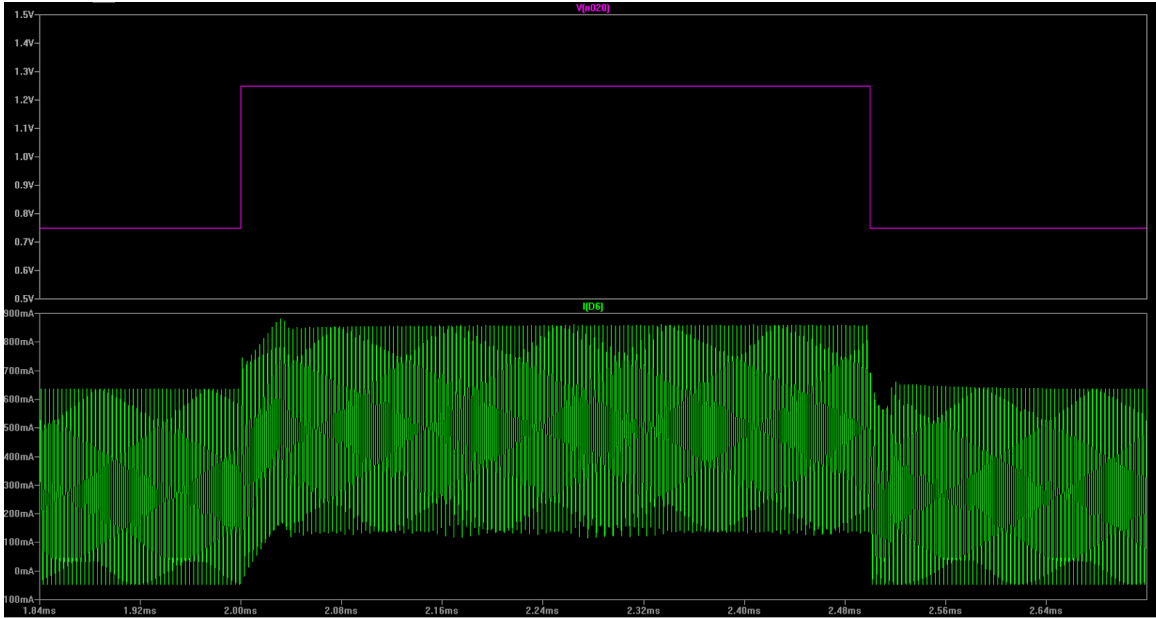
FIGURE 4-10: CTRL TEST TRANSIENT RESPONSE FOR TOPOLOGY 1

With the circuits control loop optimized, this concludes the preliminary simulation tests

for topology 1.

Topology 2: Low Price Medium Performance using LT3755 Boost then Output Cap Free

Buck

For this second topology, a single driver will drive 8 LEDs. The system

requirements for topology 2 can be seen in table 4-3.

TABLE 4-3: SYSTEM REQUIREMENTS FOR TOPOLOGY 2

V_{IN}	9-16V
V_{OUT}	0-30V
I_{OUT}	500mA
$f_{SWITCHING}$	400 kHz

Load	8 White LEDs

This system is a bit more complicated, because of the input and output voltage requirements. At times when all the LEDs are conducting, the output voltage may be above the input voltage. However, there will also be times when no LEDs are conducting and the output voltage is lower than the input voltage. Because of this, a DC/DC converter that boosts the input voltage to a certain point, then bucks it down, depending on how many LEDs are conducting, is necessary. Keith Szolusha has a patent on this topology that utilizes a single(!!) IC to achieve this. This circuit will be configured to drive 8 white LEDs at 500mA. Again, switching frequency was chosen somewhat arbitrarily based on topics discussed in chapter 3. A black box diagram of the boost then buck topology is shown in figure 4-11.
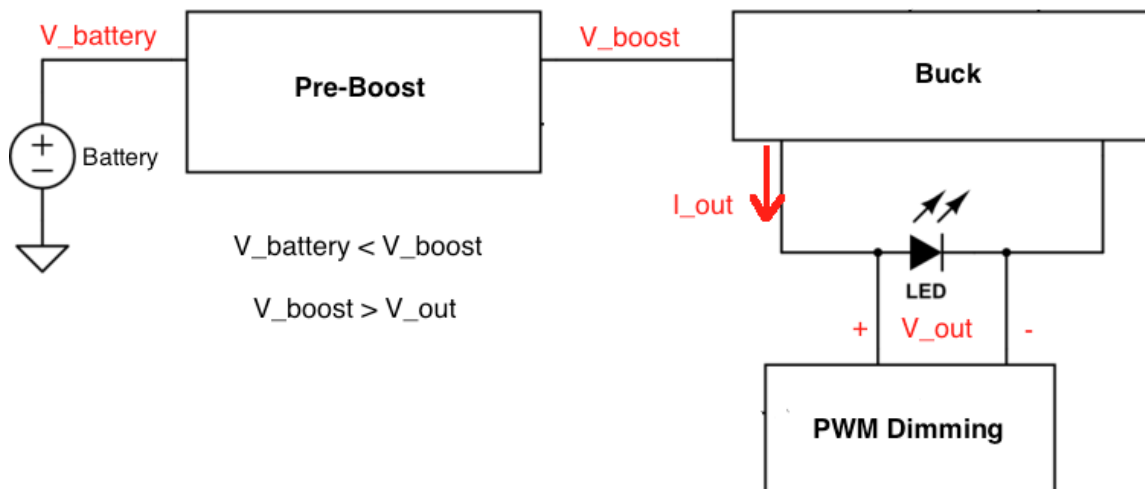
FIGURE 4-11: BOOST THEN BUCK TOPOLOGY

Since the ultimate goal is to run 12 LEDs in the system, two IC will be used to run 6 LEDs each. The LT3755 will be used in this design. This IC is capable of driving 50W loads with an input voltage range from 8-40V. This IC has similar features as the LT3797. The feedback network and current sensing follow similar rules discussed earlier in the chapter.

Before designing the boost-then-buck circuit, design of the boost and the buck will remain separate to maintain stability. Once both are designed, they can be combined for the final design.

1st Stage: Pre-Boost Circuit

As shown in figure 4-11, a pre-boost circuit will step up the battery voltage to a voltage greater than the maximum output voltage. The maximum output is dependent on the number of LEDs in the string. As mentioned earlier, each IC will run 8 LEDs for a maximum forward voltage between 24-30V, depending on the color. A reliable output voltage for the boost can be set to 40V. This circuit will maintain output voltage regulation by operating as a standard boost shown in figure 4-12.
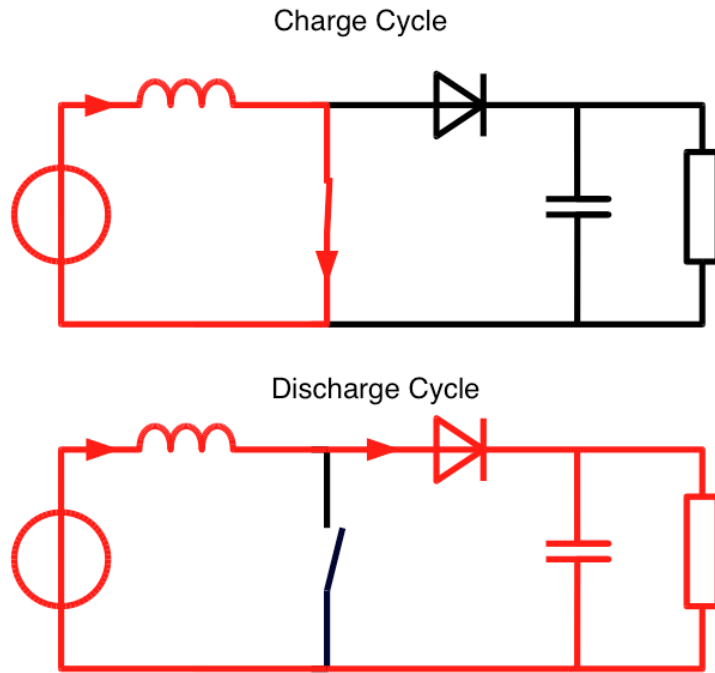
FIGURE 4-12: STANDARD BOOST CHARGING AND DISCHARGING CYCLES

When the main switch is conducting, the voltage across the inductor is equal to Vin, seen in equation 4-11.

$$V_{L\ Charging} = V_{IN}$$

(4-11)

When the switch is off, the diode is conducting, the voltage across the inductor can be seen in equation 4-12.

$$V_{L\ Discharging} = V_{IN} - V_{OUT}$$

$$where\ V_{OUT} = V_{LED}$$

(4-12)

Equation 4-13 shows the duty cycle of the standard boost circuit.

$$Duty\ Cycle = D = 1 - \frac{V_{IN}}{V_{OUT}} = 1 - \frac{12}{40} = 0.7$$

(4-13)

$$where\ V_{OUT} = V_{LED}$$

Sizing the inductor follows similar steps as circuit #1. By following equation 3-1, the inductor can be sized to 22uH with a current ripple of 300mA peak to peak. The derivation can be seen in equation 4-14.

$$\frac{V_L}{L} = \frac{dI_L}{dt} \Rightarrow L = (V_{IN}) * \frac{\Delta t_{ON}}{\Delta I_L} = \frac{(12V) * 1.75us}{700mA}$$

$$L \approx 22uH$$
(4-14)

Since this is merely a pre-boost circuit, no PWM switch is present, so just the main switch must be sized. Like before, the maximum voltage across the switch occurs during the discharging cycle when the switch is off. Equation 4-15 shows the maximum switch voltage.

$$V_{SWITCH\ (MAX)} = V_{OUT} = V_{LED} = 40V$$
(4-15)

The maximum voltage across the diode is when it is not conducting during the charging cycle. Equation 4-16 shows the maximum diode voltage.

$$V_{DIODE\ (MAX)} = V_{OUT} = V_{LED} = 40V$$
(4-16)

Along with the standard components in the boost shown in figure 4-12, output voltage feedback and peak switch current detection will be present in this design. Figure 4-13 shows the full pre-boost circuit.
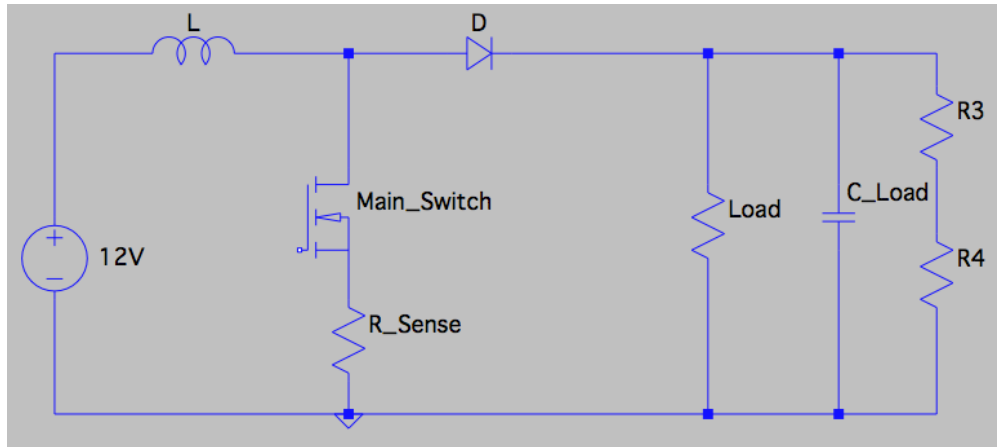
FIGURE 4-13: PREBOOST CIRCUIT FOR TOPOLOGY 2

Resistors R3 and R4 will be used for output voltage regulation. R$_{SENSE}$ will be used to maintain a safe level of peak switch current.

Feedback will be enabled by the use of a resistive divider, R3 and R4, at the output of the boost. The LT3755 has two feedback networks; one for OVP and one for output current control. The boost will utilize the OVP to maintain its output voltage. The feedback on the LT3755 is shown if figure 4-14 [8]. Equation 4-17 shows the derivation for the feedback resistors [8].
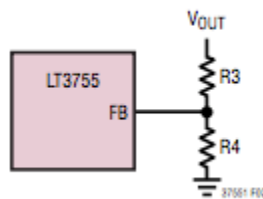
FIGURE 4-14: OUTPUT VOLTAGE REGULATION FOR PREBOOST IN TOPOLOGY 2 [8]

$$V_{OUT} = 1.25 \frac{R_3}{R_3 + R_4} = 40V \Rightarrow R3 = 1M\Omega, R4 = 32.4k\Omega$$

<div align="right">(4-17)</div>

R$_{SENSE}$ can be sized according to equation 4-18 [8]. Remember that this is the resistor needed to maintain a safe level of peak switch current.

$$R_{SENSE} < \frac{V_{IN} * 0.07}{P_{OUT}} = 21m\Omega \Rightarrow R_{SENSE} = 15m\Omega$$

<div align="right">(4-18)</div>

Now that all the components are sized, simulations can be done to ensure proper output voltage. Figure 4-15 shows the circuit.
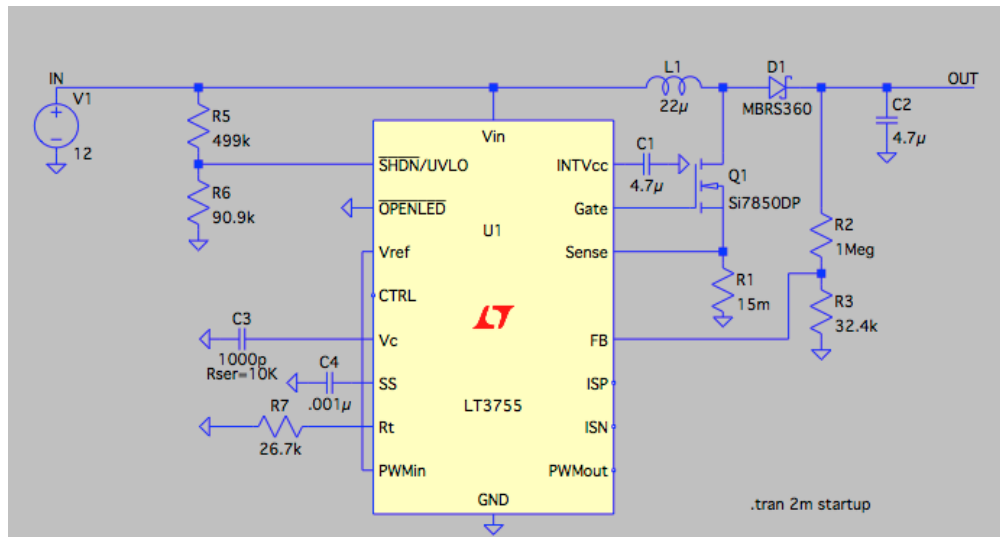
FIGURE 4-15: SCHEMATIC OF BOOST CIRCUIT FOR TOPOLOGY 2

Figure 4-16 shows the output voltage for the boost converter at the desired 43V.
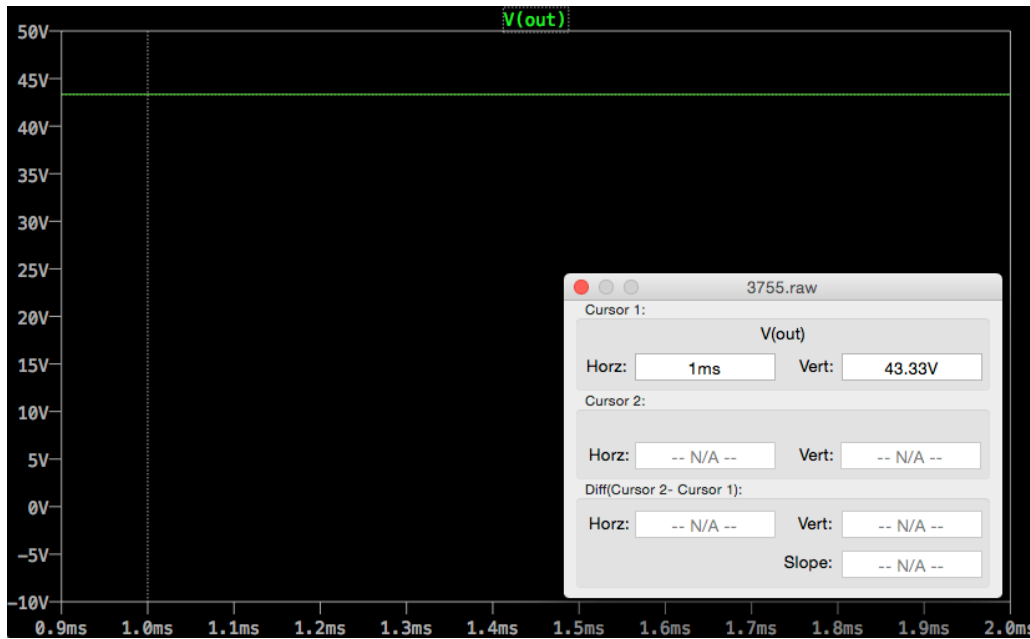
FIGURE 4-16: OUTPUT VOLTAGE OF BOOST FOR TOPOLOGY 2

The output capacitor will dictate the speed at which the circuit reacts to changes at the output of the buck. It is important that both the buck and the boost are optimized for speed. However, a buck circuit will typically have a much higher bandwidth than a boost. The boost will be the limiting factor of the overall speed of the circuit, so it important to size the output capacitor accordingly. Tests will be done on the final circuit to size this capacitor.

2nd Stage: Buck Circuit
The buck circuit used in this design is very similar to the design shown in figure XXXXX. However, the input voltage for this circuit will now be the output voltage of the boost, or 40V. The sizing of the main components (main switch, PWM switch, inductor and diode) can be seen in equations 4-19 through 4-23.

$$\frac{V_L}{L} = \frac{dI_L}{dt} \Rightarrow L = (V_{IN} - V_{OUT}) * \frac{\Delta t_{ON}}{\Delta I_L} = \frac{(40 - 24V) * 1.75us}{700mA}$$

$$\text{(4-19)}$$

$$L \approx 48uH$$

$$I_{OUT} = \frac{100mV}{R_{LED}} = 500mA \Rightarrow R_{LED} = 200m\Omega \qquad \text{(4-20)}$$

$$V_{MAIN\ SWITCH\ (MAX)} = V_{OUT\ MAX} = V_{LED\ MAX} = 30V \qquad \text{(4-21)}$$

$$V_{PWM\ SWITCH\ (MAX)} = V_{OUT\ MAX} = V_{LED\ MAX} = 30V \qquad \text{(4-22)}$$

$$V_{DIODE\ (MAX)} = V_{IN} = V_{OUT(BOOST)} = 40V \qquad \text{(4-23)}$$

The duty cycle of the buck converter is shown in equation 24.

$$Duty\ Cycle = D = \frac{V_{OUT}}{V_{IN}} = \frac{24}{40} = 0.6$$

$$\text{(4-13)}$$

$$where\ V_{OUT} = V_{LED}$$

As mentioned earlier, this design uses a single gate signal for both the boost and the buck circuit. Although the output voltage of the buck is constantly changing, the duty cycle will be changing as well. However, if you noticed, this duty cycle is very close to the duty cycle of the boot converter. It so happens that when the boost and buck circuit are operated from the same switching signal, the output voltage of the boost and output current of the buck will find some equilibrium frequency that allows the circuit to function as planned.

To improve the speed of the buck converter, the output capacitor, C5, will be removed across the load and placed across the sense resistor as shown in figure 4-17.
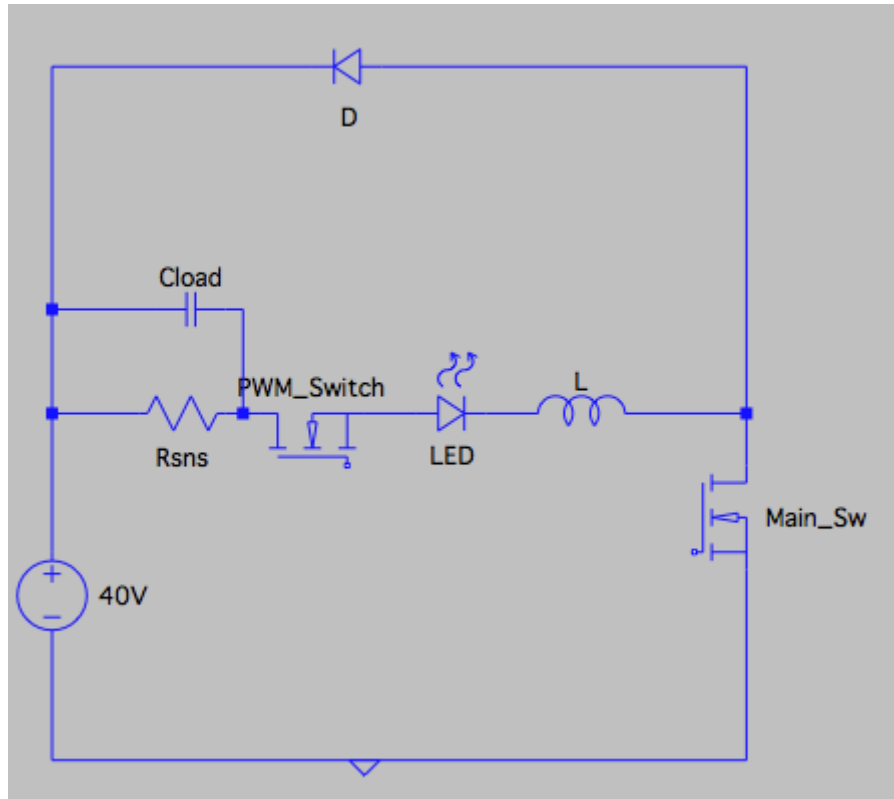
Simulations in figure 4-18 and 4-19 prove the set up for the buck circuit provides proper output current of 500mA.
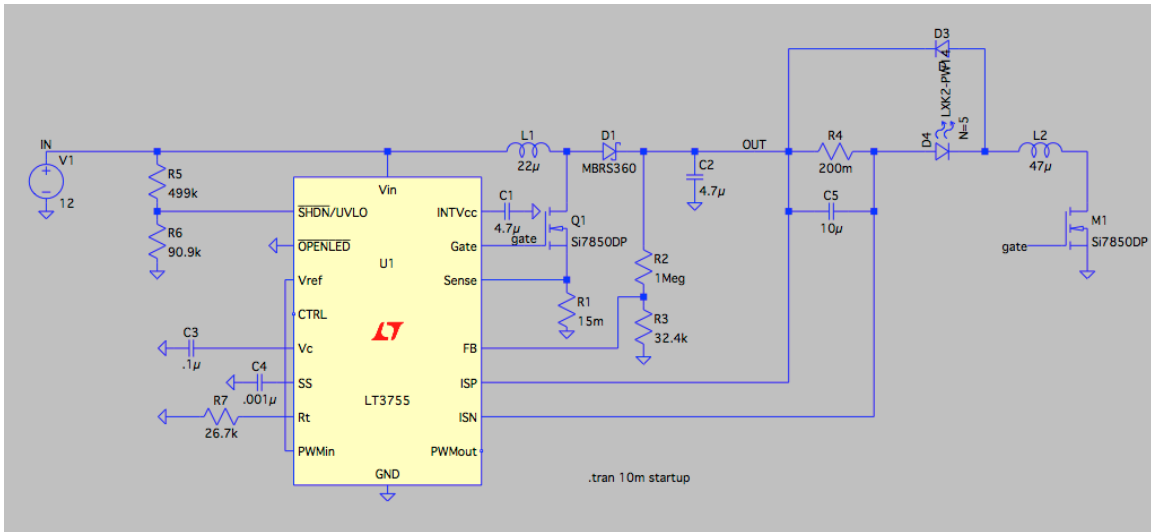
FIGURE 4-18: BOOST THEN OUTPUT CAP FREE BUCK MODE USING LT3755 FOR TOPOLOGY 2

FIGURE 4-19: OUTPUT CURRENT OF TOPOLOGY 2

To size the filter capacitor, or capacitor placed across the sense resistor, I will use the same tests used in the first topology. By altering the voltage on the Vc pin, I can simulate analog dimming. By analyzing the transient response, I can adjust the values of the RC compensation branch as well as the output capacitor to obtain a values that optimize the speed of the circuit.

Figure 4-20 shows the setup for the analog dimming test.

43
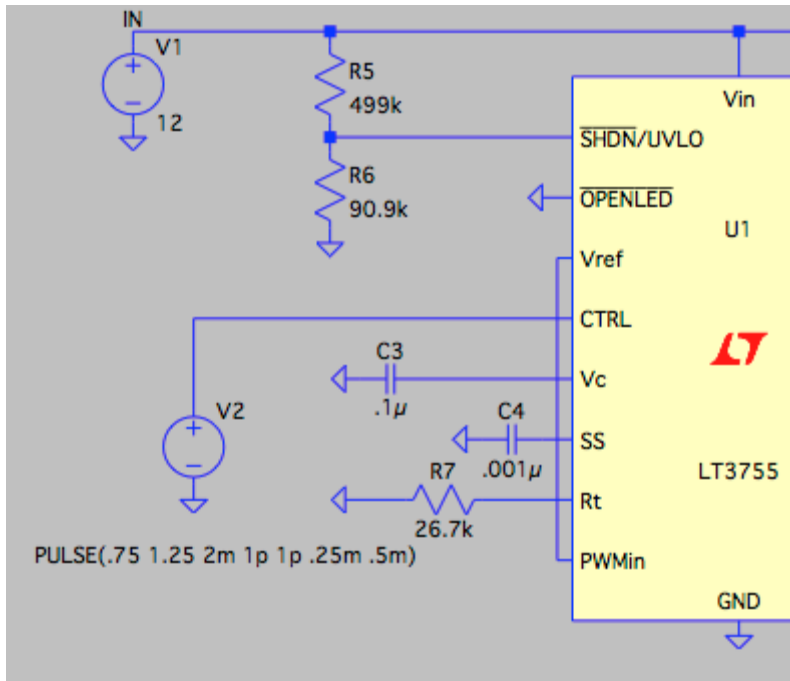
FIGURE 4-20: CTRL TEST TRANSIENT REPONSE SETUP FOR TOPOLOGY 2

The RC network and output capacitors that yielded the fastest transient response are shown in table 4-4.

TABLE 4-4: CTRL TEST COMPONENT VALUES FOR TOPOLOGY 3

Component	Value
R	5kΩ
C	3nF
C_{OUT_Boost}	4.7uF
C_{FILTER}	10nF

It is important to remember that the circuit will be driving a string of LEDs, rather than a single LED. This means that the series shunt switch used in the first topology will be insufficient. A series shunt will only allow all the LEDs to turn on or off at the same time, rather than having individual control. To work around this, a parallel shunt switch

comprised of a BJT, MOSFET, and two resistors will be connected across each LED as shown in figure 4-21. This can be called a discrete shorter outer.
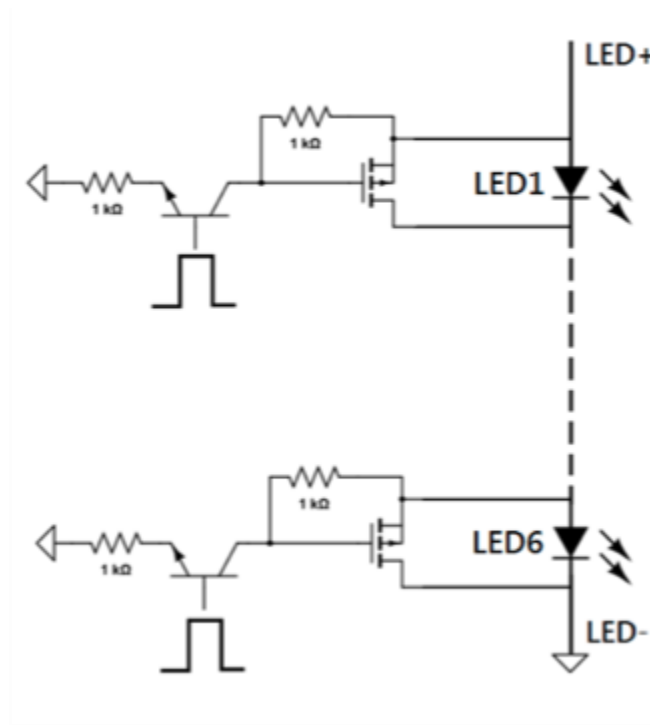
FIGURE 4-21: DISCRETE SHORTER OUTER SCHEMATIC

When the base of the BJT is pulled high, current is drawn through both resistors, creating a voltage drop across the drain and source of the MOSFET. The MOSFET turns on and current bypasses the LED through the switch. With this setup, all the LEDs can be individually controlled by a unique external PWM waveform applied to the base.

With the circuits control loop optimized, this concludes the preliminary simulation tests for topology 2.

Topology 3: Low Price Medium Performance using LT3797 Dual Boost then Output Cap

Free Buck

For this third topology, a single driver will drive 8 LEDs using the LT3797. The

system requirements for topology 3 can be seen in table 4-5.

TABLE 4-5: SYSTEM REQUIREMENTS FOR TOPOLOGY 3

V_{IN}	9-16V
V_{OUT}	0-3V
I_{OUT}	500mA
$f_{SWITCHING}$	310 kHz
Load	8 Red LED

This third topology is very similar to the second topology designed. Because of

the input and output voltage requirements, a similar set up including a pre-boost then a

buck will be needed to drive the LEDs. The LT3797 is a triple channel LED driver. By

configuring one channel as a pre-boost, the two other channels can configured as buck

circuits. In this configuration, each driver will run 8 red LEDs for a total of 16 LEDs. A

black box diagram can be seen in figure 4-22. The pre-boost steps up the input voltage up

to 30V, rather than the 40V in topology 2. Since the circuit will drive red LEDs, the

lower forward voltage results in a lower input voltage from the boost.
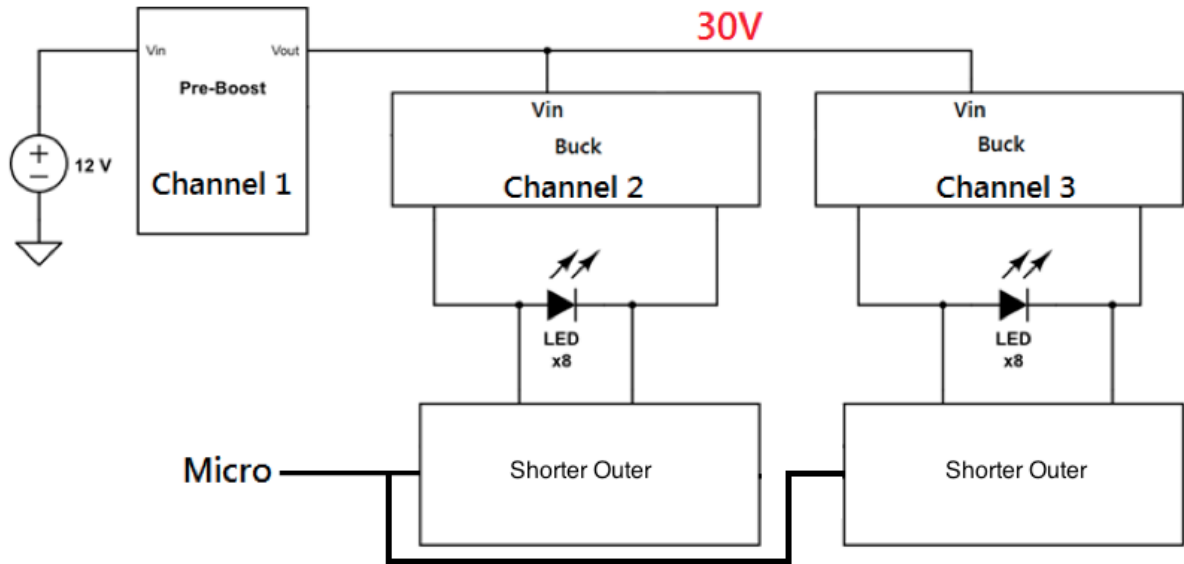
FIGURE 4-22: TOPOLOGY 3 BLACK BOX DIAGRAM

Like topology 2, I will design the boost and the buck separately and combined the two after.

1st Stage: Pre-Boost Circuit

The design procedure for this topology follows that of topology 2. The boost circuit that will be used can be seen in figure 4-23. Note that this set up is identical to the one used in topology 2.
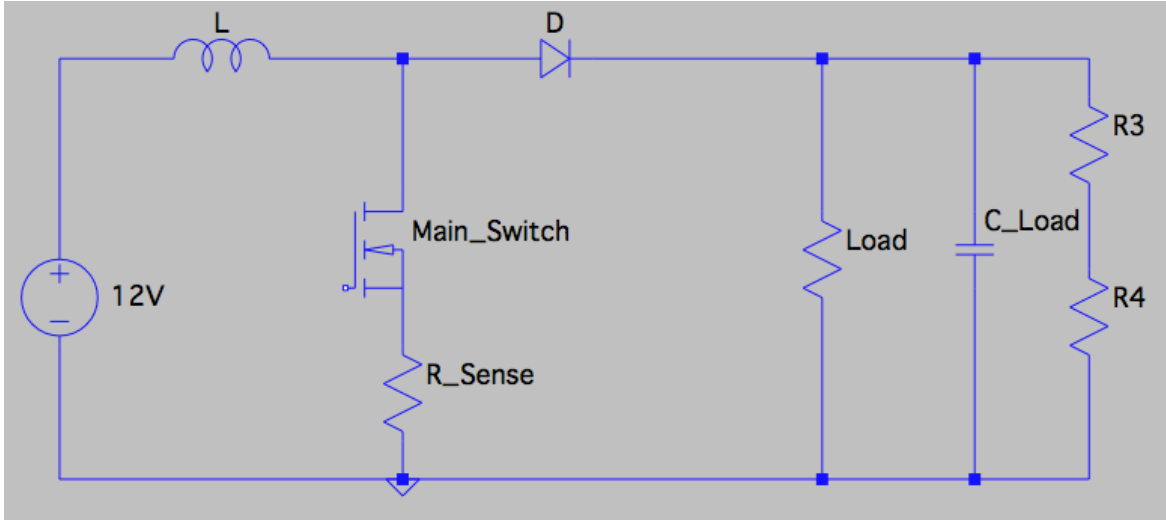
Inductor, switch, diode and sense resistor sizing can be seen in equations 4-25 through 4-28 [9].

$$V_{SWITCH\ (MAX)} = V_{OUT\ MAX} = V_{LED\ MAX} = 30V \tag{4-25}$$

$$V_{DIODE\ (MAX)} = V_{OUT\ MAX} = V_{LED\ MAX} = 30V \tag{4-26}$$

$$\frac{V_L}{L} = \frac{dI_L}{dt} \Rightarrow L = (V_{IN}) * \frac{\Delta t_{ON}}{\Delta I_L} = \frac{12V * 1.75us}{700mA}$$

$$L \approx 15uH \tag{4-27}$$

$$I_{PEAK} = \frac{80mV}{R_{SENSE}} = 4A \Rightarrow R_{SENSE} = 20m\Omega \tag{4-28}$$

Note that R_{SENSE} is the resistor that sets the peak switch current level of the boost. Since the boost will be upstream from the (two) buck converters, it is safe to assume a 2A peak switch current through each buck, totaling 4A.

The output voltage feedback for the LT3797 regulates when the voltage across R3 is 1.25V. The output voltage feedback can be sized according to equation 4-29 [9] (references according to figure 4-14):

$$V_{OUT} = 1.25 \frac{R_3}{R_3 + R_4} = 40V \Rightarrow R4 = 1M\Omega, R3 = 43.2k\Omega$$

(4-29)

To test that the pre-boost will output 30V, simulations were done is LTspice. Figure 4-24 shows the circuit diagram.
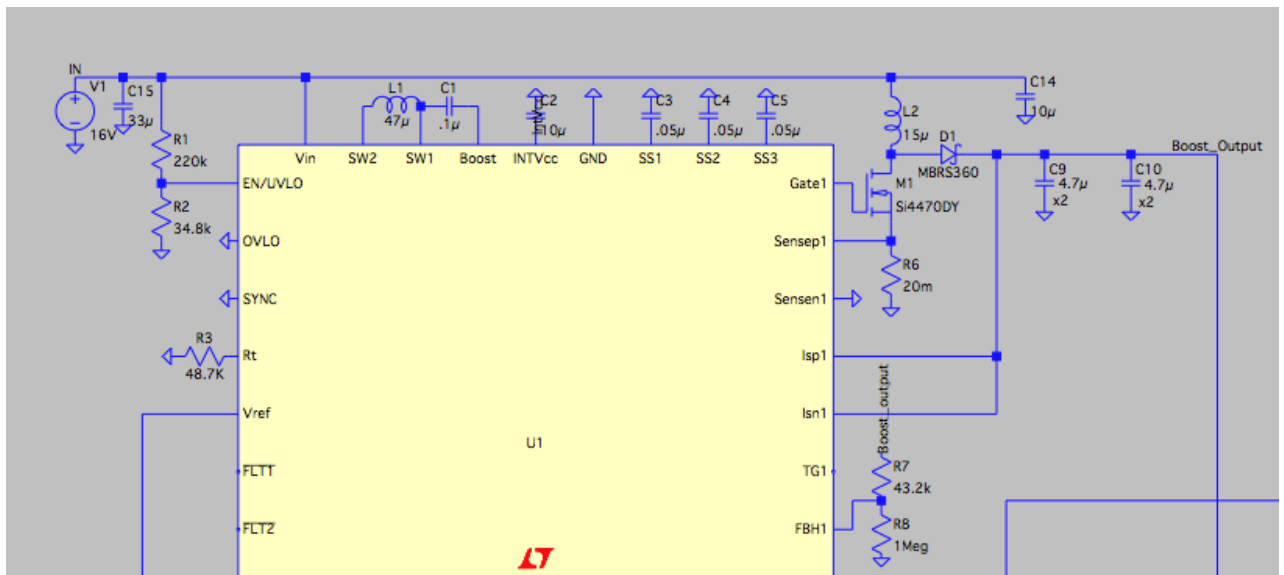
FIGURE 4-24: CHANNEL 1 OF LT3797 CONFIGURED AS BOOST

Figure 4-25 shows the output of the boost at the desired 28V.

49
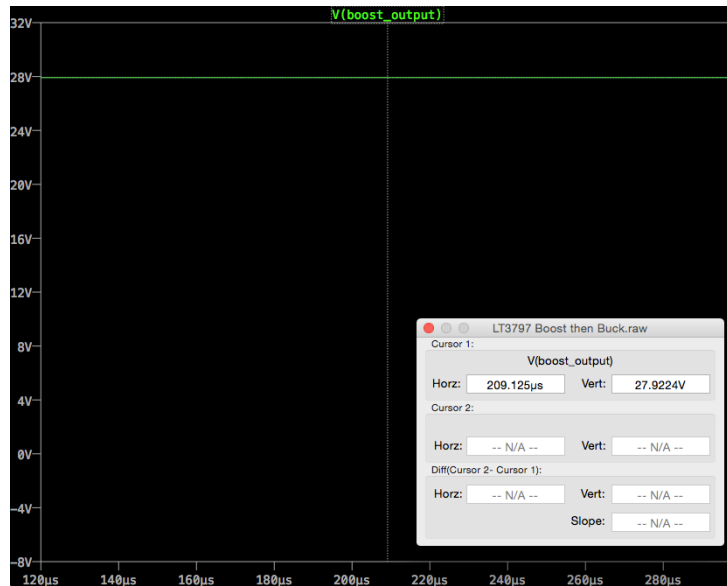
FIGURE 4-25: OUTPUT VOLTAGE OF BOOST FOR TOPOLOGY 3

2nd Stage: Buck Circuit

The 2nd stage buck circuit designed for the LT3797 will be slightly different than the one designed for the LT3755. Because the LT3797 has three separate channels, the boost and buck circuits will not share the same switching signal. This does not change the performance of the circuit, however it does change the configuration, but only slightly. The overall topology will stay the same, minus the shared switching signal. Now, each converter will have its own switching signal dictated by its feedback network.

The schematic of the circuit can be seen in figure 4-26. Note that since a string of LEDs will be attached to the load, the output capacitor is removed and placed across the sense resistor.
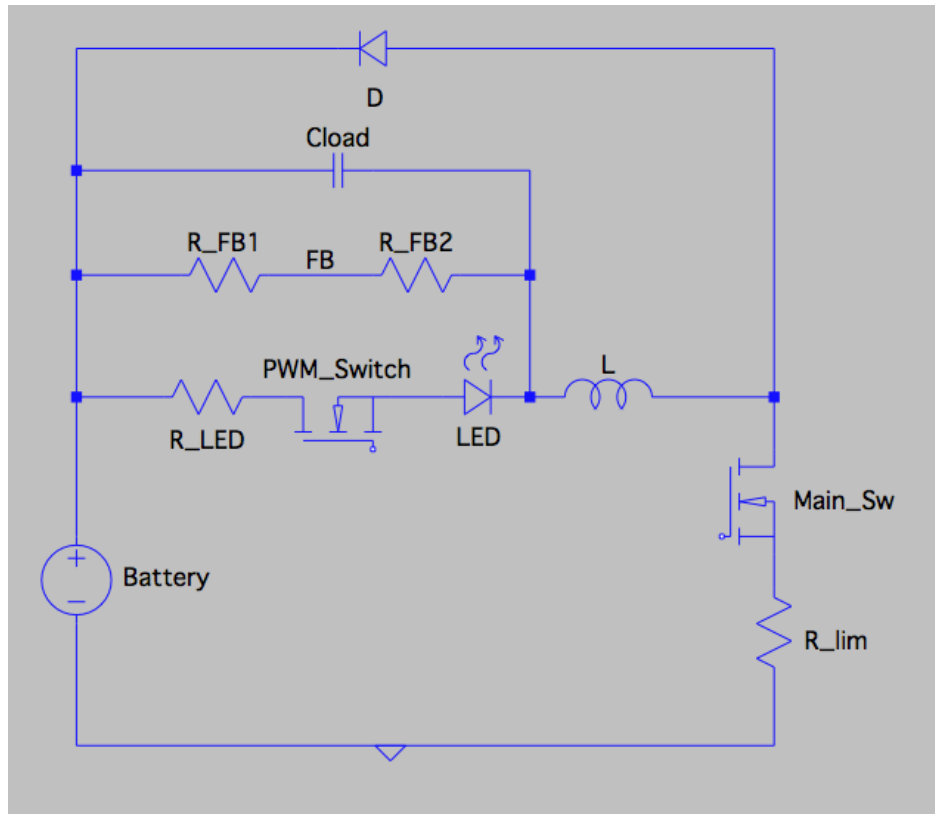
Sizing of the components follows equation 4-30 through 4-37 [9].

$$\frac{V_L}{L} = \frac{dI_L}{dt} \Rightarrow L = (V_{IN} - V_{OUT}) * \frac{\Delta t_{ON}}{\Delta I_L} = \frac{(30 - 25V) * 1.75us}{300mA}$$

(4-30)

$$L \approx 33uH$$

$$V_{MAN\ SWITCH\ (MAX)} = V_{IN} = 30V$$

(4-31)

$$V_{PWM\ SWITCH\ (MAX)} = V_{OUT\ MAX} = V_{LED\ MAX} = 25V$$

(4-32)

$$V_{DIODE\ (MAX)} = V_{IN} = 30V \tag{4-33}$$

$$I_{OUT} = \frac{250mV}{R_{LED}} = 500mA \Rightarrow R_{LED} = 500m\Omega \tag{4-34}$$

$$I_{OUT\ AVERAGE} + \frac{I_{PEAK}}{2} < I_{MAX\ PEAK} > 1A + 350mA = 1.35A \tag{4-35}$$

$$R_{SWITCH} = \frac{80mV}{I_{SWITCH\ PEAK}} = \frac{80mV}{1.6A} = 50m\Omega \tag{4-36}$$

$$V_{OUT} = 1.25\frac{R_1}{R_2 + R_1} = 25V \Rightarrow R2 = 1M\Omega, R1 = 44.2k\Omega \tag{4-37}$$

Now with the components sized, I can begin to build and integrate both circuits into one. Figure 4-27 shows the full LT3797 schematic with 2 channels configured in Buck-Mode.

Figure 4-28 shows the correct output current of 500mA through each of the LED

branches (green = channel 2, teal = channel 3) and the output voltage of the boost(blue).
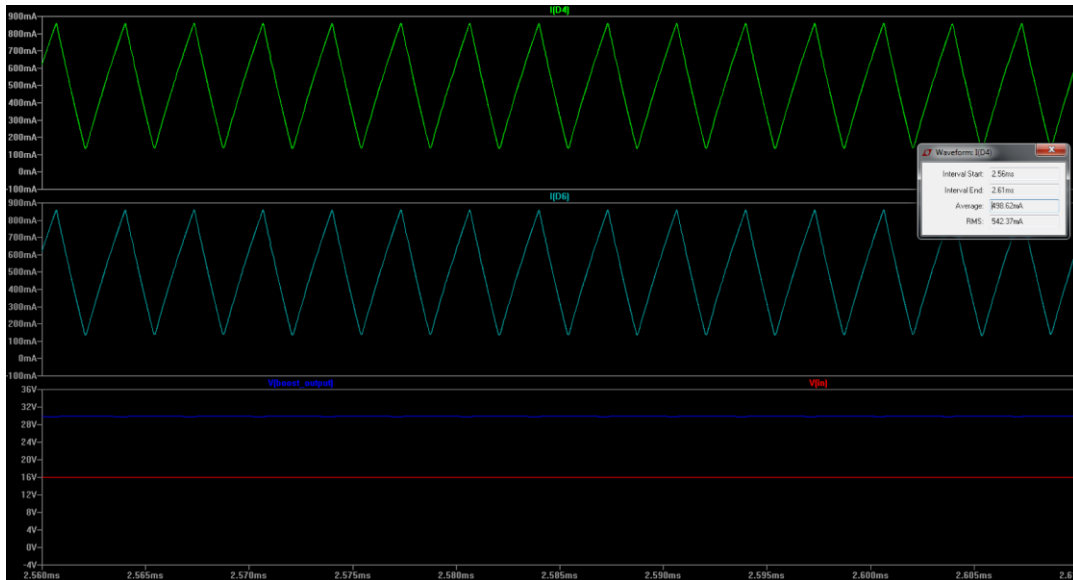
The next test optimizes the circuits speed. As mentioned in the previous designs, by altering the voltage on the Vc pin of the IC, analog dimming can be achieved. When analog dimming is in effect, the transient response can be analyzed to obtain the fastest response with the best R, C and output capacitor combination. Figure 4-29 shows the set up for this test.
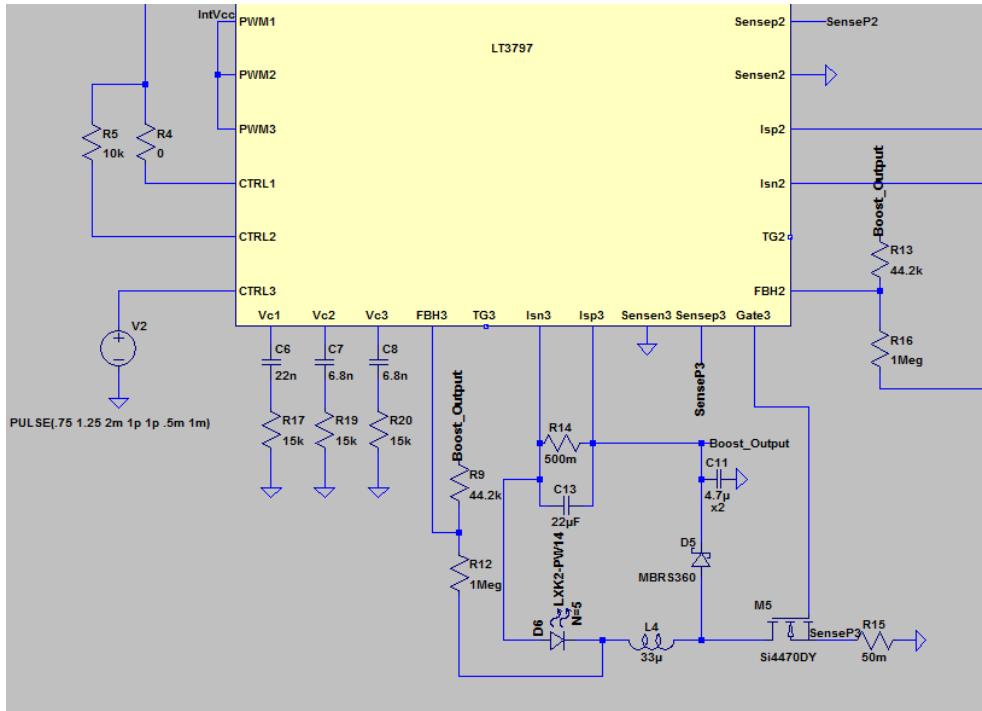
FIGURE 4-29: CTRL TEST SETUP ON CHANNEL 3 FOR TOPOLOGY 3

The RC network and output capacitors that resulted in the fastest transient response are shown in table 4-6.

TABLE 4-6: CTRL TEST COMPONENT VALUES FOR TOPOLOGY 3

Component	Boost Values	Buck Values
R	5kΩ	5kΩ
C	22nF	6.8nF
C_{OUT}	10uF	22uF

With the circuits control loop optimized, this concludes the preliminary simulation tests for topology 3.

Note that during the design and simulation phase, microcontroller design and implementation was left out. This will be covered in chapter 5 when hardware tests can be run simultaneously to verify proper operation. Also, due to the nature of these circuits regulation component, tests including line and load regulation were omitted.

Chapter 5 : Hardware Results

In this chapter, the physical implementation of the design specifications discussed in chapters 3 and 4 will be implemented. Component and board selections for the LED drivers will be chosen. Hardware verification as well as bode plot analysis and transient response tests will also be presented. Once the hardware is completed, the PWM waveforms generated from the microcontroller will be integrated with the drivers to have a fully functional integrated system that achieves adaptive brake lighting.

5.1. Component/Board Selection

Topology 1: LT3797 configured as Buck Mode

Demo board DC1784A will be used to implement the buck converter using the LT3797. The DC1784A was originally designed to run the LT3797 with 3 independent boost channels. However, during the design of this demo board, considerations were made to allow minor changes to the board so that it could be configured to run 3 independent buck channels. By making a few cuts on the board, as well as removal/addition of certain components, the circuit shown in Figure 5-1 was set up. Figure 5-1 shows a single buck channel of the LT3797 for simplicity. All three channels were configured identically.
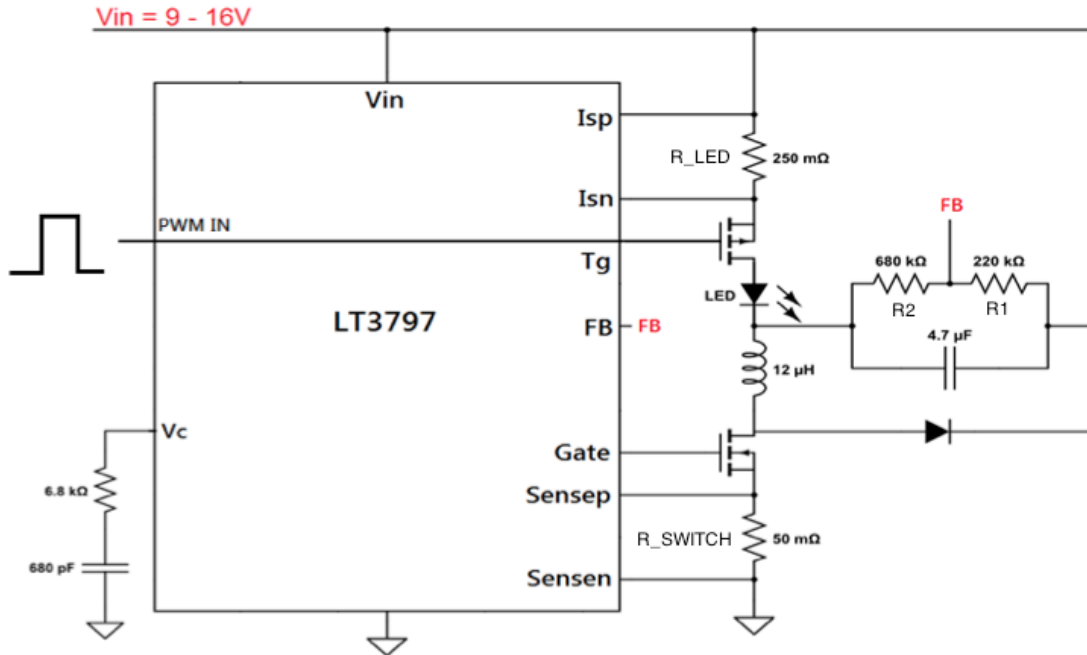
FIGURE 5-1: INITIAL HARDWARE CIRCUIT DIAGRAM FOR TOPOLOGY 1

Component selection follows the procedures outlined in Chapter 4. Table 5-1 lists the main components chosen for the design on the DC1784A. All resistors and capacitors used were standard value surface mount 5% (unless otherwise specified) rated that varied in size from 1210 – 0402.

TABLE 5-1: COMPONENT SELECTIONS FOR TOPOLOGY 1

Name	Ratings	Manufacturer	Part Number
Main Switch	60V 19A	Infineon	BSC039N06NS
PWM Switch	100V 3A	Vishay	SI7113DN
Inductor	12uH	Wurth	744777112
Diode	60V 3A	Diodes Inc.	PDS360
R_LED	250mΩ 1%	-	-
R_SWITCH	50mΩ 1%	-	-

Topology 2: LT3755 Boost then Output Cap Free Buck

     Topology 2 will be implemented on the demo board designed for the LT3755, DC1268B-B. Similar to the first topology, to implement the boost then buck, small cuts and the removal/addition components were made to the DC1268B-B. The schematic in Figure 5-2 shows the circuit built.
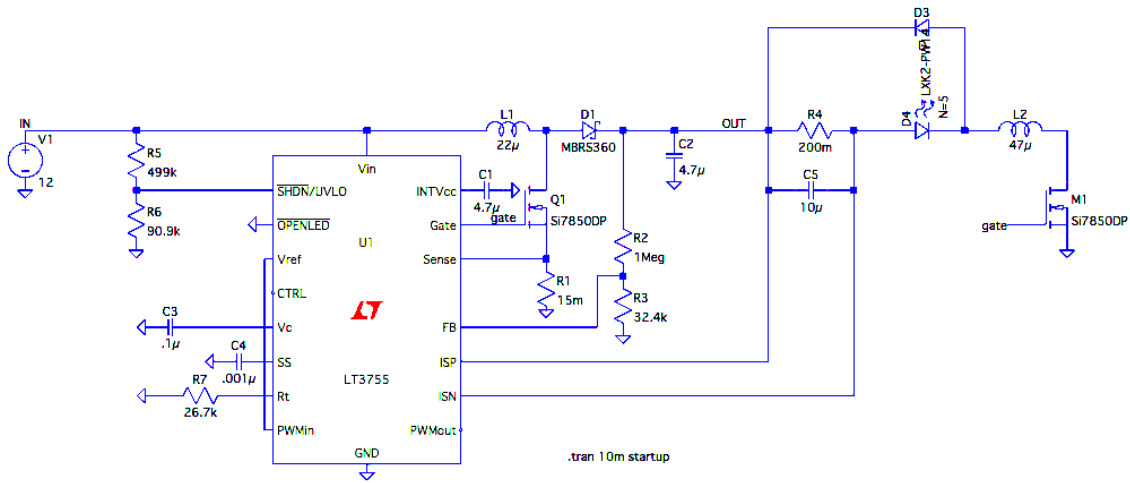
FIGURE 5-2: INITIAL HARDWARE DESIGN SCHEMATIC FOR TOPOLOGY 2

     Component selection follows the procedures outlined in Chapter 4. Table 5-2 lists the main components chosen for topology 2. All resistors and capacitors used were standard value surface mount 5% (unless otherwise specified) rated that varied in size from 1210 – 0402.

TABLE 5-2: COMPONENT SELECTIONS FOR TOPOLOGY 2

Name	Ratings	Manufacturer	Part Number

Boost Switch	60V 3A	Vishay	Si7850DP-T1
Buck Switch	60V 3A	Vishay	Si7850DP-T1
Boost Inductor	22uH	Wurth	744065220
Buck Inductor	47uH	Wurth	744066470
Boost Diode	60V 5A	Diodes Inc.	PDS560-13
Buck Diode	60V 5A	Diodes Inc.	PDS560-13
R1	15mΩ 1%	-	-
R4	200mΩ 1%	-	-

Discrete Shorter Outer:

The schematic used for the shorter outer is shown in figure 5-3.
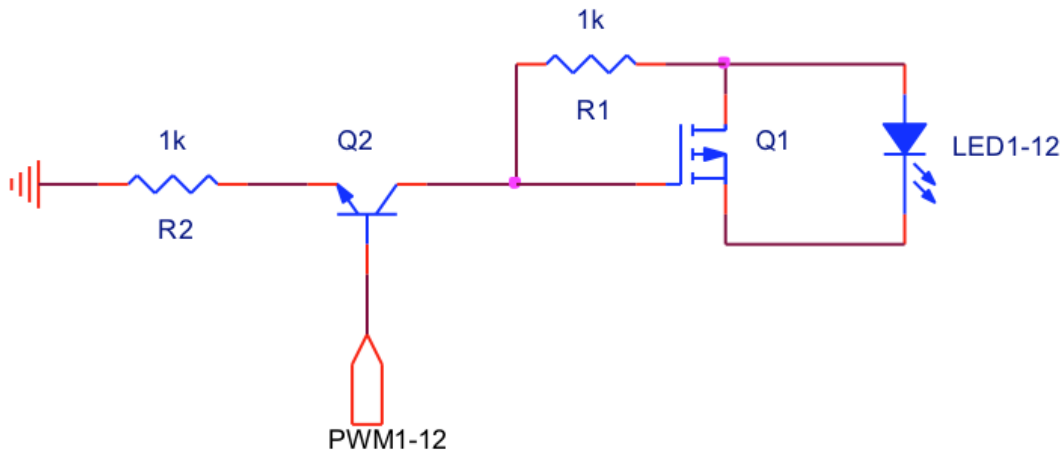
Remember this circuit is necessary to individually PWM dim LEDs configured in a string. By pulling the base of Q1 high, Q1 turns on and current is drawn through the resistors, creating a voltage across V_{DS} of Q2. This creates a parallel shunt switch across the LED, turning the LED off. Table 5-3 shows the components used for this design.

TABLE 5-3: COMPONENT SELECTIONS FOR DISCRETE SHORTER OUTER

Name	Rating	Manufacturer	Part Number

Q1	60V 200mA	ON Semi	2N3904
Q2	60V 1A	Vishay	Si2309

Topology 3: LT3797 Dual Boost then Buck

Due to the flexible nature of the DC1784A, similar alterations were made to the

board to convert two of the original boost circuits to buck converters. By simply

connecting the output of the pre-boost to the input of the buck with an external wire, the

circuit was created in hardware as depicted in Figure 5-4.
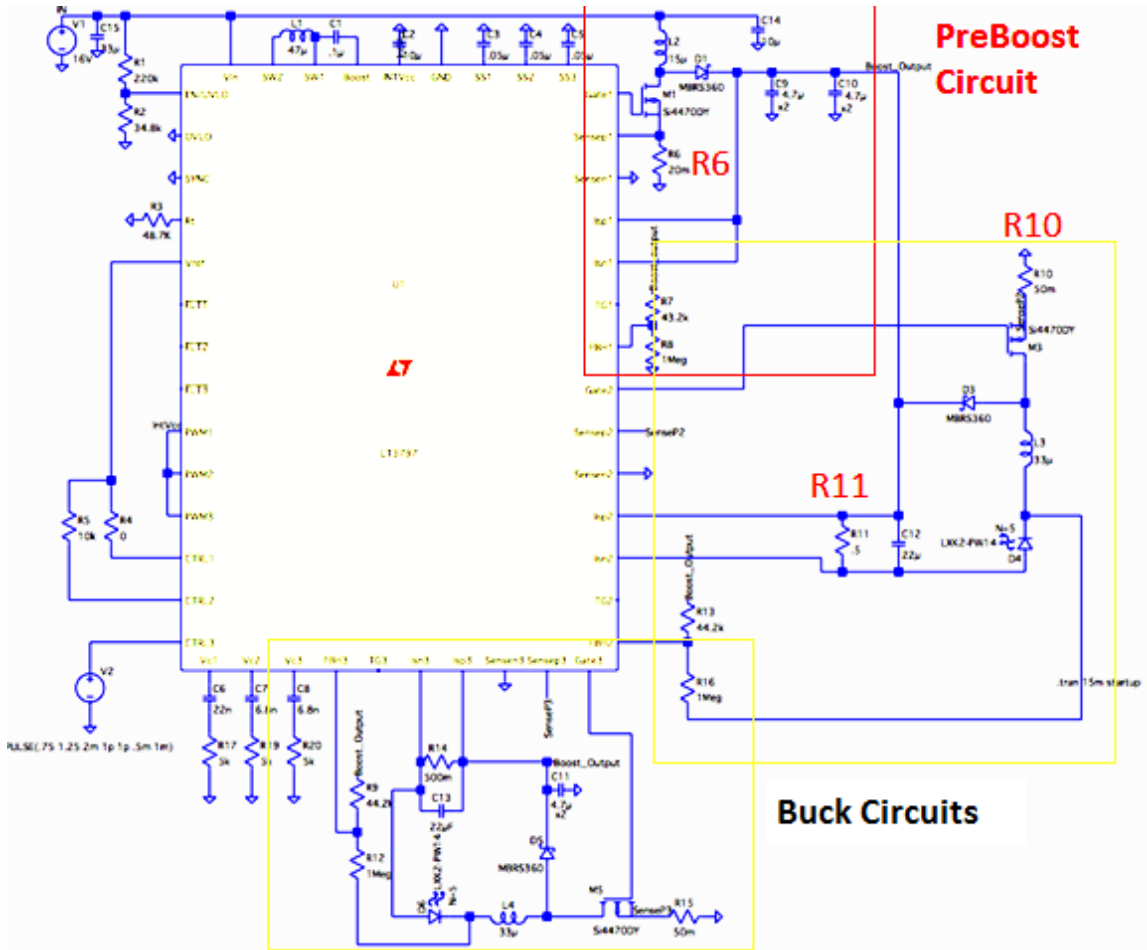
FIGURE 5-4: INITIAL HARDWARE SCHEMATIC DESIGN FOR TOPOLOGY 3

Table 5-4 shows the component selection for topology 3. Note that channel 2 and 3 of the LT3797 are designed to be identical. R10 and R11 in table 5-3 correspond to R15 and R14, respectively.

Name	Ratings	Manufacturer	Part Number
Boost Switch	60V 19A	Infineon	BSC039N06NS
Buck Switch	60V 19A	Infineon	BSC039N06NS
Boost Inductor	15uH	Wurth	744065220
Buck Inductor	33uH	Wurth	744066470
Boost Diode	60V 3A	Diodes Inc.	PDS360
Buck Diode	60V 3A	Diodes Inc.	PDS360
R6	20mΩ 1%	-	-
R10	50mΩ 1%	-	-
R11	500mΩ 1%	-	-

Discrete Shorter Outer:

The discrete shorter outer used in topology 3 is identical to the one used in topology 2. The schematic and component selection can be seen in figure 5-3 and table 5-3.

5.2. Output Current Tests

The first test checks the output current regulation and makes sure the appropriate output current is being produced. Figures 5-5 show the output current for topology 1.
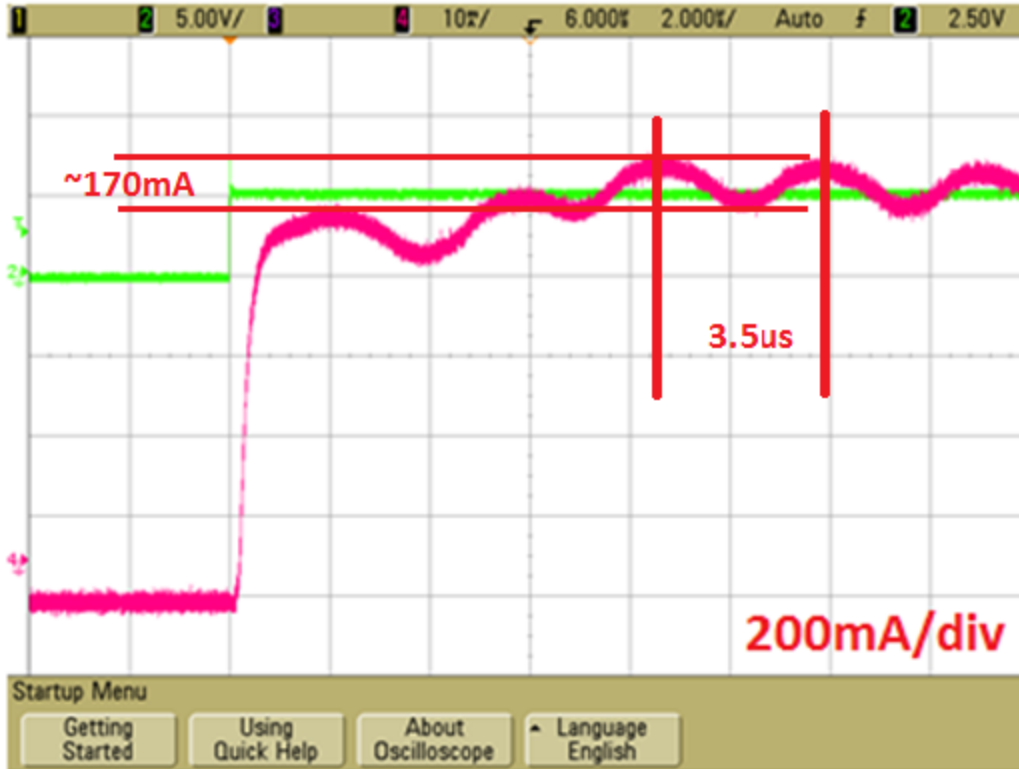
FIGURE 5-5: OUTPUT CURRENT FOR TOPOLOGY 1

As expected, topology 1 outputs a current of 1A. The output current waveform is relatively flat, with a peak-to-peak ripple of 170mA. The inductor current has a peak-to-peak ripple of 700mA. Due to the output capacitor placed across the LED, most of the AC ripple present in the inductor is absorbed and not present in the load current. The ripple in the load current has a period of 3.5us and a frequency of 300 kHz, roughly the switching frequency of the converter. Channels 1, 2 and 3 output currents were verified to be 1A. A summary of the output current can be seen in table 5-5.

Similar tests were executed for topologies 2 and 3 to ensure output current was regulated at the specified amount. However, it is important to note that the current waveforms look different than topology 1. This is due to the removal of the output capacitor across the load. Since there is no output capacitor, there is nothing to absorb the

63

AC ripple present in the inductor. Now, the inductor current is equal to the load current and shares the same ripple. It may seem unusual that the ripple on the output current is so high. One might speculate that a large ripple current through the LEDs will distort the brightness, maybe making it flicker. The switching frequency of the converter is much higher than the threshold that a human eye can distinguish the current switching through the LEDs. Another key aspect of topology 2 and 3 is the configuration of the load. The output voltage will constantly be changing when PWM dimming the LEDs. This will change the amount of ripple through the inductor and the load. According to equation XXX, as the voltage across an inductor increases, so does the change in current. For topologies 2 and 3, when all LEDs are conducting, or at full load, output voltage is at its maximum. This means the ripple will be at its largest. At no load, or when no LEDs are conducting, the output voltage is small and determined by the on resistance of the transistors in the discrete shorter outer. The ripple is not as important at no load. What is important is that the converter is still outputting its regulated current, but this will be discussed later on. Table 5-5 summarizes the output currents for topologies 2 and 3.

TABLE 5-5: SUMMARY OF OUTPUT CURRENTS

Topology Number	Output Current @ Full Load	Output Current Ripple(peak to peak)	Output Voltage @ Full Load
1	1A	170mA	2.7V
2	500mA	700mA	24V
3	500mA	650mA	21.6V

Once output currents were regulated to the desired amount, the next tests were performed.

5.3. Transient Response Tests

Once the circuit is outputting the proper current, next is to optimize the circuit for speed. Speed of the circuit is how fast the converter will correct the output to the regulated amount when a change occurs at the input or load. To understand how to speed up the control loop, understanding how the control loop operates is necessary.

For example, the LT3797 has two different feedback networks that allow output current regulation. First, there is the sense resistor in which the IC monitors the voltage across this resistor. Error amplifiers and comparators will adjust the duty cycle of the main switch if the voltage across the sense resistor varies from its set value of 100mV. This ensures that the proper amount of current (1A in the case of topology 1) is delivered to the load. The second feedback loop is the overvoltage control. This monitors the voltage across the load to ensure it does not exceed the value set by the feedback resistors. The voltage feedback is the second line of defense in this system. For example, imagine the transition from the LED conducting (in topology 1) to the series shunt switch turning on, cutting current off to the load. Initially, the converter is regulating 1A to the output. However, once the PWM switch turns off, current is then cut from the load. Since no current is flowing, the sense resistor has no voltage across it. The converter will try to correct this and shove more and more current into the load. This is where the second feedback loop comes into play. Since the PWM switch is open, the voltage across the load and feedback network increase enough to trigger the overvoltage control. It is important to set the feedback resistors correctly so this sequence occurs.

This feedback network is even more important in topologies 2 and 3 due to the configuration of the load. Both the LT3797 and LT3755 share very similar feedback networks as they both monitor the voltage across the sense resistor and voltage across the feedback resistors. The loads in these topologies (2 and 3) are configured in a string. During transitions from X LEDs on to Y LEDs on (where X and Y are an integer number), the output voltage will be constantly changing, triggering the voltage feedback to change the duty cycle of the converter. The converter will think the change in output voltage is due to a lack of or excess current through the load. The converter will try to make up for this by either cutting or supplying current to the load. However, the current level was already at an appropriate level and this will result in possibly unsafe levels of current through the remaining conducting LEDs. The sense resistor feedback network regulates this change in output current to the necessary level. The speed at which the circuit responds to the load change will determine how large the current surge will be. This is why a fast control loop is necessary.

Adjusting the speed of the control loop for the LT3797 and LT3755 is very easy. As mentioned in chapter 4, there is a V_C pin where a resistor and capacitor are placed in series to ground. Figure 5-6 shows the V_C pin locations for channel 1, 2 and 3 for the LT3797 (similar to the LT3755). The size of the resistor and capacitor will determine how fast the control loop reacts. A larger resistor will create a voltage across it more quickly than a smaller valued resistor. A larger capacitor will allow more current to flow into this branch. Alternatively, a smaller capacitor allows less current through the branch. Secondly, the output capacitor has a substantial role, as this limits how fast the output voltage can change. The speed at which the voltage across the resistor changes (due to the nominal

values of the RC circuit), as well as the output capacitor, will dictate how fast the control loop reacts.
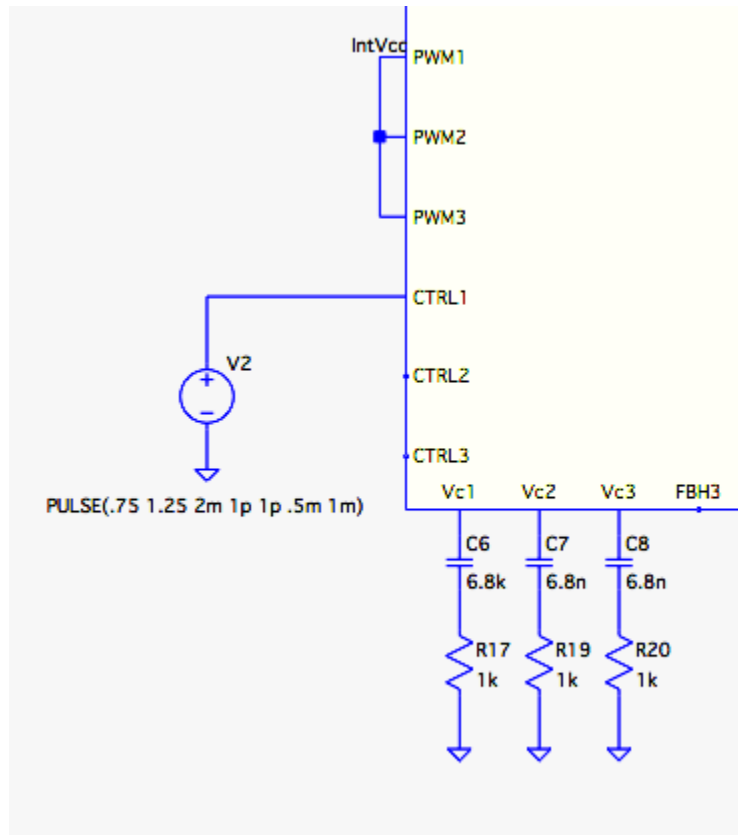
FIGURE 5-6: TOPOLOGY 1 CTRL T TEST SETUP

The first test is identical to the tests done in the simulation and design phase discussed in Chapter 4. By changing the voltage on the CTRL pin of the IC, the output current levels will change from 50% to 100% of the final value. By tweaking the RC compensation on the V_C pin, as well as the output capacitor (for both buck and boost if applicable), the transient response will become closer to a critically damped system. Figure 5-6 shows the test set up. The voltage generator present in figure 5-6 was implemented with a function

generator. The frequency of the generator was changed based on how well the circuit responded to the change in output current. Performing this test in hardware takes into account the parasitic inductance and capacitance within the circuit that was not accounted for in the simulations. By using an RC board, with multiple resistor and capacitor values that can be changed simply by a switch makes this test much easier to complete. Table 5-6 lists the CTRL and V_C pin number for all three topologies. Since the LT3797 is a triple channel device, the pin locations for each channel are listed in order.

TABLE 5-6: CTRL AND Vc PIN LOCATIONS

IC	$V_c(1,2,3)$ Pin Location	CTRL(1,2,3) Pin Location
LT3797	19,28,40	8,9,10
LT3755	15	16

In summary, the previous test will be done with different RC values connected to the Vc pin as well as varying output capacitor values. The configuration that yields the most critically damped waveform will be listed.

Input voltage was set to 12V for all three topologies. Output current was set to 1A for topology 1 and 500mA for 2 and 3.

Topology 1:
Characteristics of the transient response that result in a "fast" response is a steep rising and falling edge, while maintaining a critically damped waveform. An underdamped system may reach its final value quicker, but also comes with a surge of unwanted current through the load. Figure 5-7 shows the transient response of a single channel in response to a 70μs pulse. The following R, C and C_{OUT} values are shown in table 5-7.
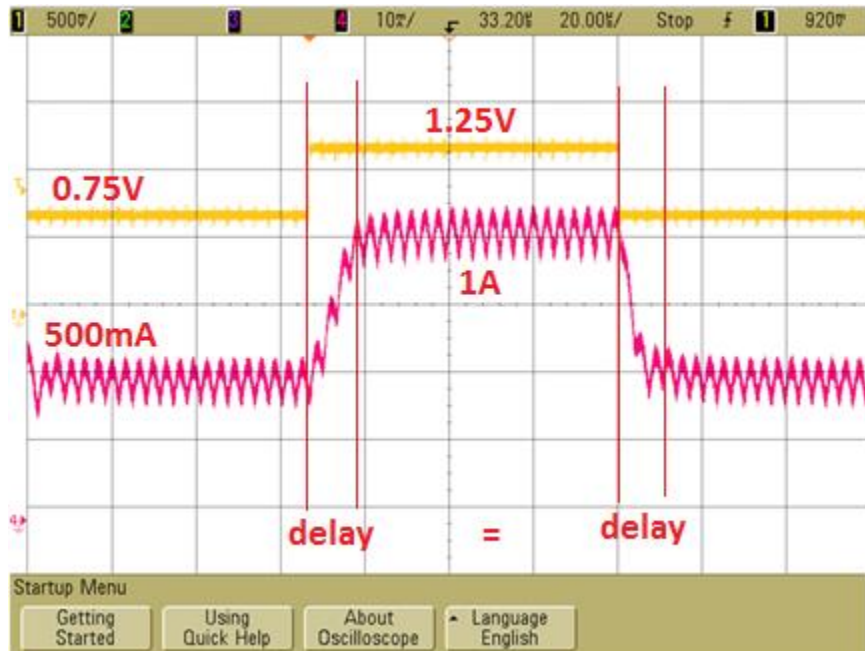
FIGURE 5-7: CTRL TEST TRANSIENT REPONSE FOR TOPOLOGY 1

Figure 5-7 shows the CTRL voltage (yellow) rise from 0.75V to 1.25V. This tells the converter to change the output current from 50% to 100% of its final value. The R, C and C_{OUT} values used to obtain figure 5-7 are shown in table 5-7. Note that the time delay on the rising edge is equal to the time delay on the falling edge. This indicates the amount of time the converter takes to change its output current from 50% to 100%.

TABLE 5-7: COMPONENT VALUES FOR TOPOLOGY 1 AFTER CTRL TEST IN HARDWARE

Component	Value
V_C Resistor	3.5kΩ
V_C Capacitor	750pF
Output Capacitor	4.7uF

With the values shown in table 5-7, transient tests are concluded for topology 1.

The first transient response test is done for topology 2 compares the performance of a circuit with an output capacitor across the load versus the capacitor placed across the sense resistor. It is important to gain qualitative data that proves whether one topology is better performance-wise than another. This test involves a transition from X LEDs on to Y LEDs where X and Y represent a number of LEDs conducting at a given time. Because of the load configuration, a surge of current will occur during this transition. Each transition from X LEDs to Y LEDs was performed to ensure that all possible transitions operated safely. This transition was chosen somewhat arbitrarily just to show the surge of current. Most transitions resulted in a very similar surge. Figure 5-8 and 5-9 shows the comparison of the transient response for the circuit with and without the capacitor in a transition from 6 LEDs to 5 LEDs on. This test was done using the discrete shorter outer designed in chapter 4 and implemented with the PIC microcontroller.
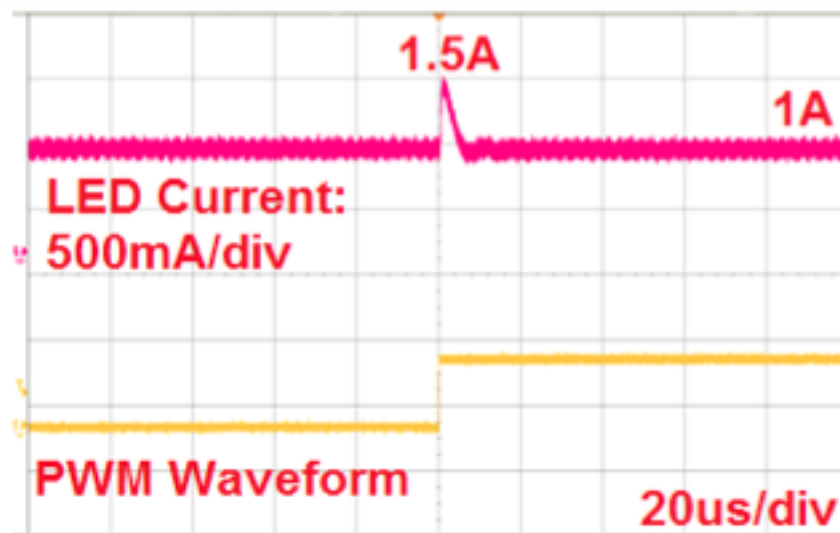
FIGURE 5-8: TRANSIENT RESPONSE WITH OUTPUT CAP ACROSS LOAD IN TOPOLOGY 2

The circuit with the output capacitor results in a large surge of output current, up to 50% its original value. This is unsafe and can damage the load or the device. The circuit without the output capacitor has a slight disturbance in the current, but is able to respond quickly to maintain proper output current. Bode plot tests were run on the buck stage to obtain quantitative measurements about the two circuits. This means that the boost circuit was not hooked up while conducting this test. This is important because the numbers obtained in table 5-8 are not the final circuit's performance figures. Section 5.4 will discuss how these plots were obtained. Table 5-8 summarizes the findings.

TABLE 5-8: QUANTITATIVE COMPARISON OF OUTPUT CAPACITOR LOCATION

Measurement	WITH Output Capacitor	Without Output Capacitor
Crossover Frequency [kHz]	50.4	119.3
Phase Margin [deg]	69.51	63.1
Gain Margin [dB]	10.64	12.95

Quantitative data proves that the output cap free circuit is a much faster alternative.

Table 5-9 shows the component values that yielded the results in table 5-9. Remember that since this topology runs off of a single IC, the V_C RC network is the same. However, the preboost and the buck have an output capacitor that will help dictate the speed of the circuit.

TABLE 5-9: COMPONENT VALUES FOR TOPOLOGY 2 AFTER CTRL TEST IN HARDWARE

Component	Value
V_C Resistor	20kΩ
V_C Capacitor	680pF
Output Capacitor of Boost	4.7uF
Output Capacitor of Buck	22uF

Next, the CTRL test was performed on the Boost then Output Cap Free Buck circuit, shown in figure 4-17 with the component values shown in table 5-9. Figure 5-10 shows the transient response. Because of the unique configuration of topology 2, the pre-boost circuit limits the bandwidth and speed of the entire converter. This slow response results in an underdamped system and an overshoot of 170mA.
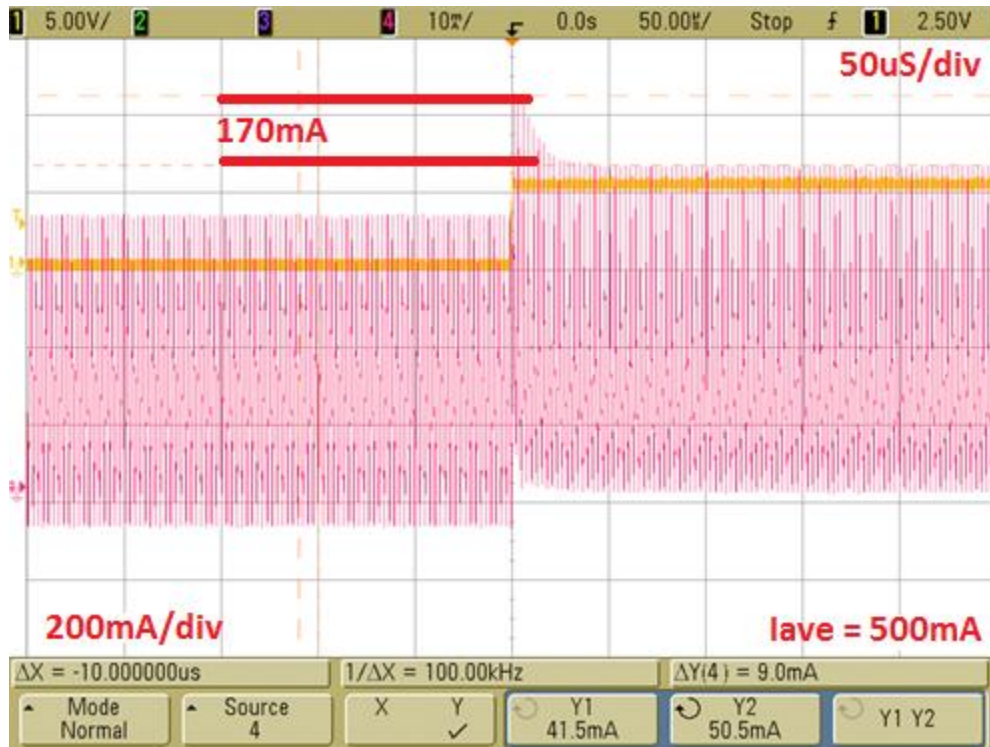
FIGURE 5-10: CTRL TEST TRANSIENT RESPONSE FOR TOPOLOGY 2 (WITH OUTPUT CAP REMOVED)

With the values shown in table 5-9, transient tests are concluded for topology 2.

Topology 3:

      Similar tests were ran on topology 3 that concludes removal of the output

capacitor across the load results in a faster control loop and transient response. Table 5-

10 shows the test set up that resulted in the fastest transient response. Since the LT3797 is

a triple channel LED driver, the boost and buck circuits will have their own V_C pin, R, C

and C_{OUT} values.

TABLE 5-10: COMPONENT VALUES FOR TOPOLOGY 3 AFTER CTRL TEST IN HARDWARE

Components	Boost Values	Buck Values
V_C Resistor	20kΩ	23kΩ
V_C Capacitor	10nF	4.7nF
C_{OUT}	4.7uF	10uF

Next, the second transient analysis test was performed to analyze the various transitions with the test set up shown above. Similar to topology 2, the unique configuration results in a small surge of current and an underdamped system. However, this small amount will not affect circuit operation of light intensity.

With the values shown in table 5-10, transient tests are concluded for topology 3.

5.4. Bode Plot Analysis

After completing the transient analysis tests, the control loops for the circuits are almost optimized. However, the previous test only gave qualitative data rather than quantitative data. It would be ideal to have numerical values to prove that the circuit is fast and stable. This can be done through bode plot analysis. To maintain stability in a DC/DC converter, it is recommended to have a gain margin (GM) less than |10dB| and a phase margin of 60 deg or greater. These values are margins that dictate the phase and magnitude of the feedback loop. To small of a phase margin or too large of a gain margin will result in improper operation of the feedback loop, leading to instability. Tweaking the R,C (connected to the V_C pin) and C_{out} values, like the previous test, will shift the crossover frequency to give a numerical value of how fast the circuit actually is. Crossover frequency is directly related to how "fast" the circuit responds to a change in input or load values. The goal is to obtain a combination of the resistor and capacitor values that results in the highest crossover frequency, while still remaining stable.

To obtain these bode plots, a small injection signal of 10-100mV must be applied to some node in the feedback network. A small valued resistor will aide the injection

signal into making a voltage difference in the feedback network. When the injection

signal is applied, a frequency response analysis tool (in my case, the Venable) will

measure the phase and magnitude difference of two points (channel 1 and channel 2) in

the feedback loop due to the disturbance of the injection signal. This phase and

magnitude difference of channel 1 and channel 2 will indicate the phase margin (PM) and

gain margin (GM) of the set up. By injecting a signal at the ISN node of each topology,

as shown in figure 5-11, the magnitude and frequency response can be obtained. All tests

were performed with an input voltage of 12V.
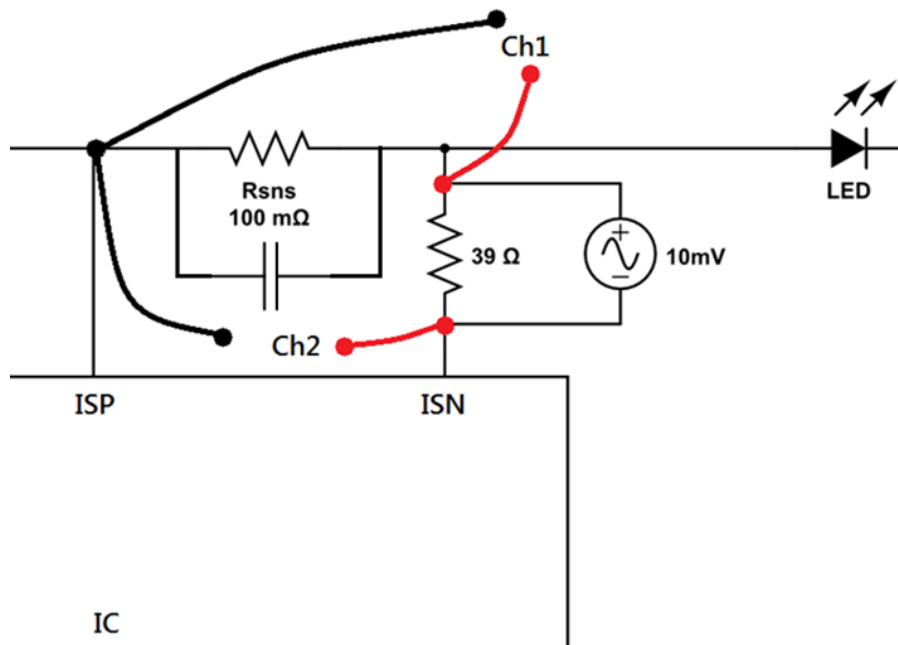
FIGURE 5-11: INJECTION SIGNAL DIAGRAM FOR OBTAINING BODE PLOT MEASUREMENTS USING

VENABLE

75

Figure 5-12 shows the bode plot that resulted in the fastest crossover frequency.

Table 5-11 summarizes the values obtained in figure 5-12. Table 5-12 lists the component

values used to obtain the plot.
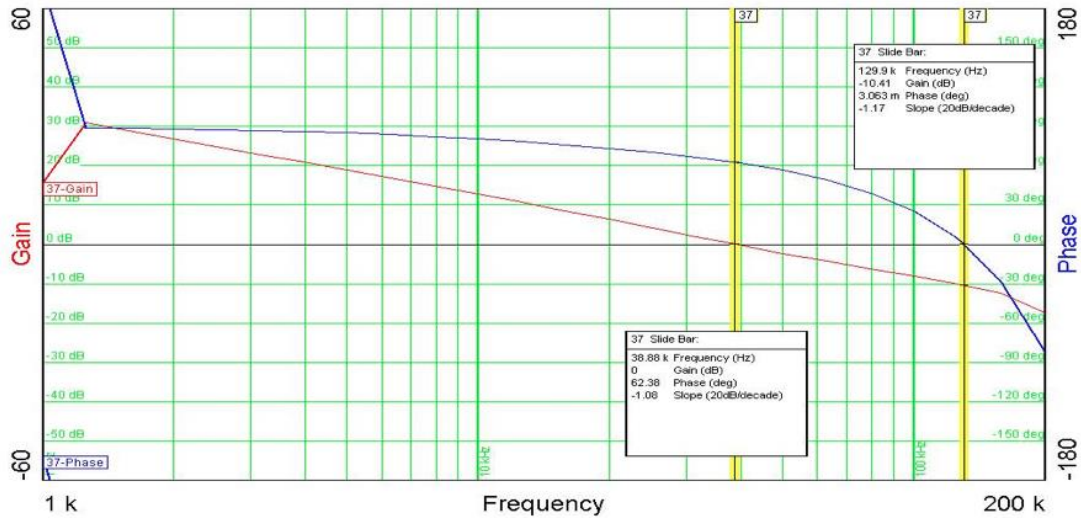
FIGURE 5-12: BODE PLOT FOR TOPOLOGY 1

TABLE 5-11: BODE PLOT MEASUREMENTS FOR TOPOLOGY 1

Measurement	Values
Crossover Frequency[kHz]	38.88
PM[deg]	62.39
GM[dB]	10.41

TABLE 5-12: TOPOLOGY 1 COMPONENT VALUES USED FOR BODE PLOT TEST

Component	Values
V_C Resistor	6.8kΩ
V_C Capacitor	6.8nF
C_{OUT}	4.7uF

Using the component values shown in table 5-12 yielded the results shown in table 5-11. This concludes the control loop optimization for topology 1.

Topology 2:

For topology 2, the identical injection signal and resistor were applied to the ISN node of the feedback network. Similar steps were done to obtain the crossover frequency, gain and phase margin. Table 5-13 summarizes the performance characteristics that yielded the largest crossover frequency. Table 5-14 summarizes the component values used to obtain those measurements

TABLE 5-13: BODE PLOT MEASUREMENTS FOR TOPOLOGY 2

Measurement	Values
Crossover Frequency [kHz]	5.1
PM [deg]	81.29
GM [dB]	11.27

TABLE 5-14: TOPOLOGY 2 COMPONENT VALUES AFTER BODE PLOT TEST

Component	Values
V_C Resistor	10kΩ
V_C Capacitor	2.2nF
Boost C_{OUT}	4.7uF
Buck C_{OUT}	10uF

Note that the crossover frequency of topology 2 is much smaller than topology 1. This is the first evidence that proves that topology 1 is a higher performance option compared to topology 2. This concludes the control loop optimization for topology 2.

Topology 3:

   Again, for topology 3, the identical injection signal and resistor were applied to

the ISN node of the feedback network. Similar steps were done to obtain the crossover

frequency, gain and phase margin. Table 5-15 summarizes the performance

characteristics that yielded the largest crossover frequency. Table 5-16 summarizes the

component values used to obtain those measurements

TABLE 5-15: BODE PLOT MEASUREMENTS FOR TOPOLOGY 3

Measurement	Values
Crossover Frequency [kHz]	9.8
PM [deg]	69.2
GM [dB]	11.17

TABLE 5-16: TOPOLOGY 3 COMPONENT VALUES AFTER BODE PLOT TEST

Components	Boost Values	Buck Values
V_C Resistor	15kΩ	15kΩ
V_C Capacitor	22nF	6.8nF
C_{OUT}	4.7uF	22uF

Topology 3 is slightly faster than topology 2, but not by much. It is, however, much

slower than topology 1, proving that topology 1 is the highest performance topology.

With the information obtained from the above tables, this concludes the control loop

optimization for topology 3.

5.5. PWM Verification

   Now that the circuits are optimized for speed and stability, PWM applications can

begin. This is where the fun begins, as the lights can be programmed for different

patterns to simulate adaptive brake or head lighting systems. First, qualitative measurements must be made on each topology to compare speed. This means obtaining the highest PWM dimming ratio for each topology. Remember that PWM dimming ratio is the period of PWM pulses divided by the on-time of the pulse. Figure 5-13 shows an example calculation of PWM dimming ratio.
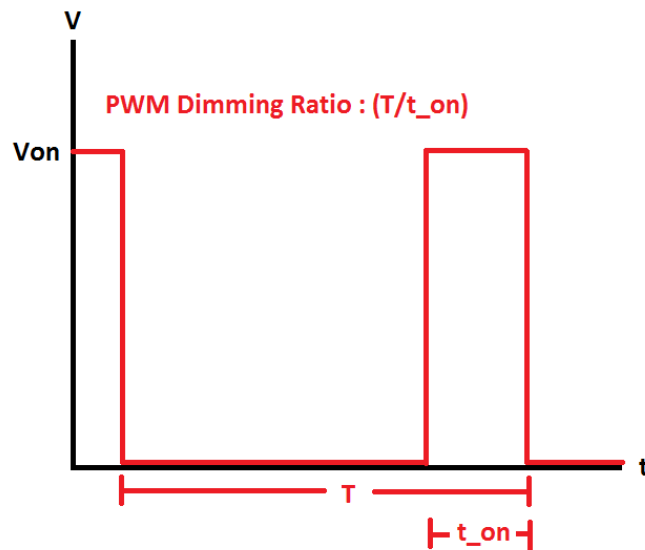
FIGURE 5-13: PWM DIMMING RATIO EXAMPLE

By following the design requirements specified in chapter 3, the microcontroller, PICLF877A, will be integrated with the converters. This microcontroller has 16 GPIO pins and a 10 bit ADC. Commands written in C generate square waveforms that vary in pattern based on an analog input read from the ADC. Various patterns were made to showcase the different styles that can be achieved, given a specific shape of a tail/head light. The code can be found in the appendix. The 12 outputs of the PIC microcontroller were routed to the corresponding PWM pins of the IC or the shorter outer. Topology 1 routes these external PWM signals straight to the ICs PWM pins of the LT3797.

Topologies 2 and 3 route the external PWM signals to the discrete shorter outer that control the parallel shunt switches across each LED.

Topology 1:
To obtain the PWM dimming ratio for topology 1, the input voltage was set to a median value between 9 and 16V, or 12V. Output current was at its full value of 1A. The PWM frequency was set to 100Hz (T = 10ms). The on-time of a single LED was changed from 1ms to 1us and the transient waveform was observed. For a certain PWM Dimming ratio to be "approved", the waveform had to reach its final value within the on-time of the pulse. Choosing the lowest value is almost subjective to the engineering observing the waveforms, as reaching the final value could have a couple different meaning. Some may argue that reaching its final value for as little as 0.5us would qualify as sufficient, but some may argue a longer steady state final value must occur.

Figure 5-14 shows the PWM waveform and transient response to a 2us pulse. This results in a **PWM Dimming Ratio of 5000:1**. As the picture shows, the current (pink) reaches its final value of 1A in 1us and stays at the final value for 1us. This is indicative of a good transient waveform. 5000:1 Dimming ratio is a very high ratio, indicating the quickness of the control loops response to a change at the output. This is overkill for a system involving adaptive brake and headlights, as these systems would only require something within the 100:1 dimming range. This high performance circuit could be used for adaptive lighting in an airplane, where the ambient light is so low, a very dim light is sufficient for the pilot to see his instruments.
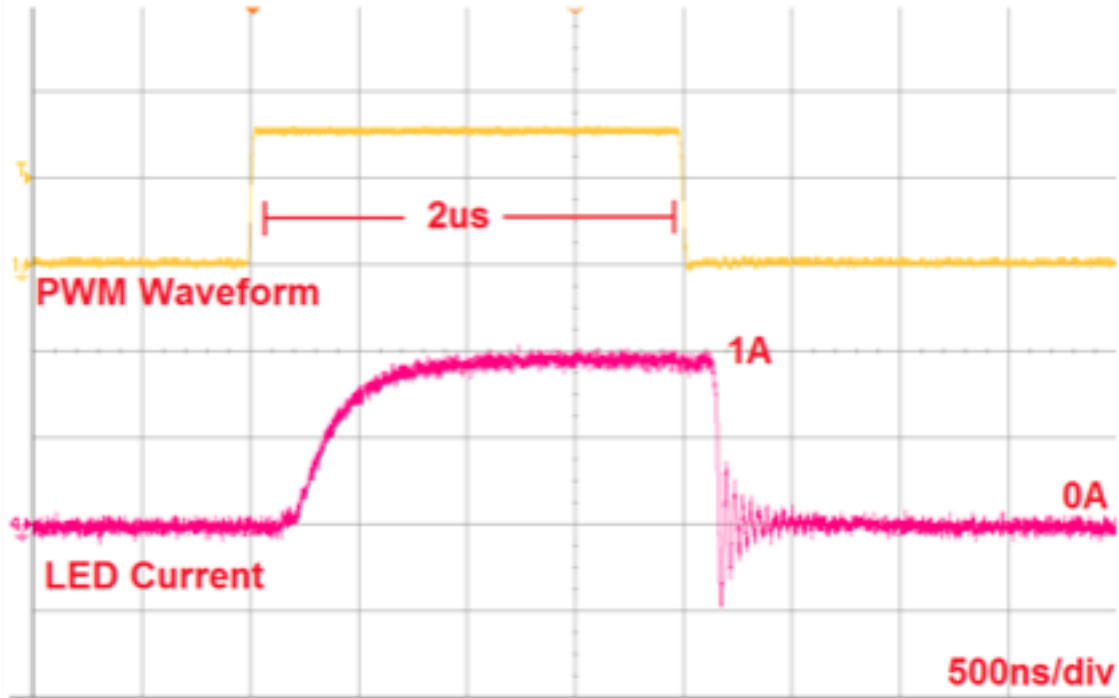
FIGURE 5-14: PWM DIMMING RATIO OF 5000:1 FOR TOPOLOGY 1

Topology 2:

Obtaining the PWM Dimming ratios for topology 2 is slightly different than before. Because the load is configured as a string, rather than a single LED, it is expected that a lower dimming ratio will be obtained. Instead of PWM dimming a single LED, a single parallel shunt switch will be used to turn on and off all the LEDs in a string at a time. With an input voltage of 12V, output current of 500mA, and PWM switching frequency of 100Hz, the test was run. Table 5-17 summarizes the data.

TABLE 5-17: PWM DIMMING RATIO OF TOPOLOGY 2

PWM Dimming Ratio	500:1

With a PWM switching frequency of 100Hz, a 500:1 dimming ratio translates to a 20us pulse. This means it takes 10 times longer to reach its final output current value than topology 1.

Topology 3:
> For topology 3, the same test set up from topology 2 was used to obtain the information present in table 5-17.

TABLE 5-18: PWM DIMMING RATIO OF TOPOLOGY 3

PWM Dimming Ratio	750:1

> With a PWM switching frequency of 100Hz, a 500:1 dimming ratio translates to a 13us pulse. This means it takes 6 times longer to reach its final output current value than topology 1. Topology 3 is slightly faster than topology 2. Given the PWM dimming ratios, applications involving head or brake light would be ideal for topologies 2 and 3.

> With the first topology, each converter drives a single LED. The transient response is limited to two options, the LED turning on or off. This is a simple case in which the transient response only affects a single LED. However, this was not the case for the 2nd and 3rd topology as the load is a string of LEDs. As mentioned in earlier chapters, when an LED turn off or on, it changes the output voltage of the converter, therefore introducing a surge of current through the load. If too many LEDs turn off or on at a certain point, an unwanted surge of current could damage the IC or the load. To avoid this, phase shifted PWM dimming was enabled. This means that all the LEDs would turn off in sequential order to avoid the high surge of current. This was done through software. A transient response of zero LEDs on to all the LEDs on can be seen in

figure 5-15. During each transition, the LED current subsides and reaches its steady state value before the next transition begins.
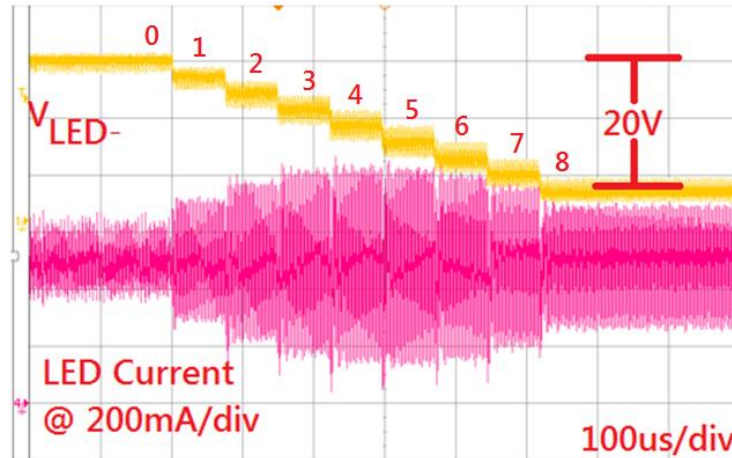
FIGURE 5-15: PHASE SHIFTED PWM DIMMING

Another key transition to pay attention to is when switching from some LEDs on, to no LEDs on. As mentioned in chapter 2, when no LEDs are conducting, the output voltage of the converter is close to zero (or the series resistance of the MOSFETs internal resistances). Each IC has a minimum on time it must meet. If this minimum on time is not met, the converter will shut off current to the output. This is not ideal. When no LEDs are conducting, current should be at its full value so when an LEDs turns back on, it does not have to wait a certain time for the inductor to charge back up to the final current value. However, the on-resistance of the MOSFETs in the shorter outer created enough voltage so that the minimum on-time was met when all LEDs were off. This allows the inductor current to maintain continuity and be ready to supply current once an LED turns back on. The transient response from 1 LED on to all off can be seen in figure 5-16. Both topology 2 and 3 experience the same transient waveform. The arrow represents the transition from 1 LED on to all LEDs off.
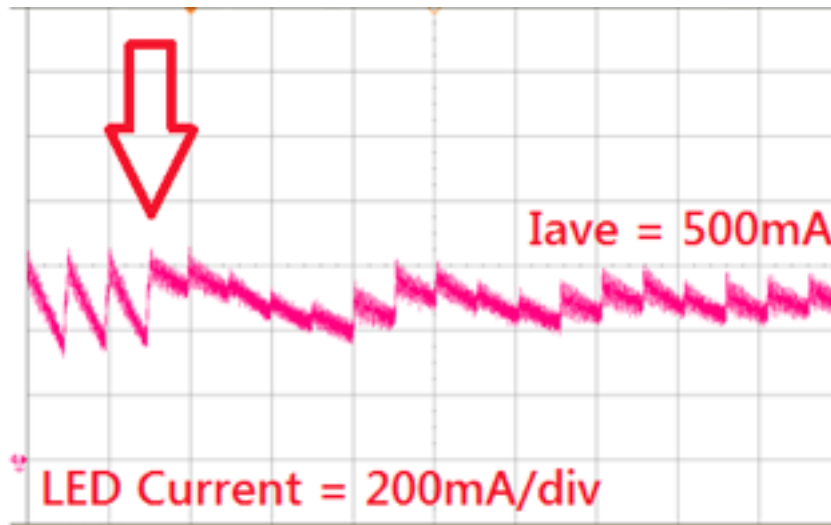
FIGURE 5-16: TRANSIENT REPONSE FROM 1 LED ON TO 0 LEDS ON

All of the different load transitions were analyzed and deemed safe for operation. This concludes the PWM verification for topologies 1, 2 and 3.

5.6. Final Design

Now, the analog converters are integrated with the digital PWM waveforms, completing the hardware verification of the adaptive brake light systems. With time to spare, I created a number of various light patterns to further showcase the abilities of the converters.

The final design schematic for topology 1 can be seen in figure 5-17. This includes all the final component values as well as the components listed in section 5.1. Figure 5-18 and 5-19 shows a picture of the final hardware layout.
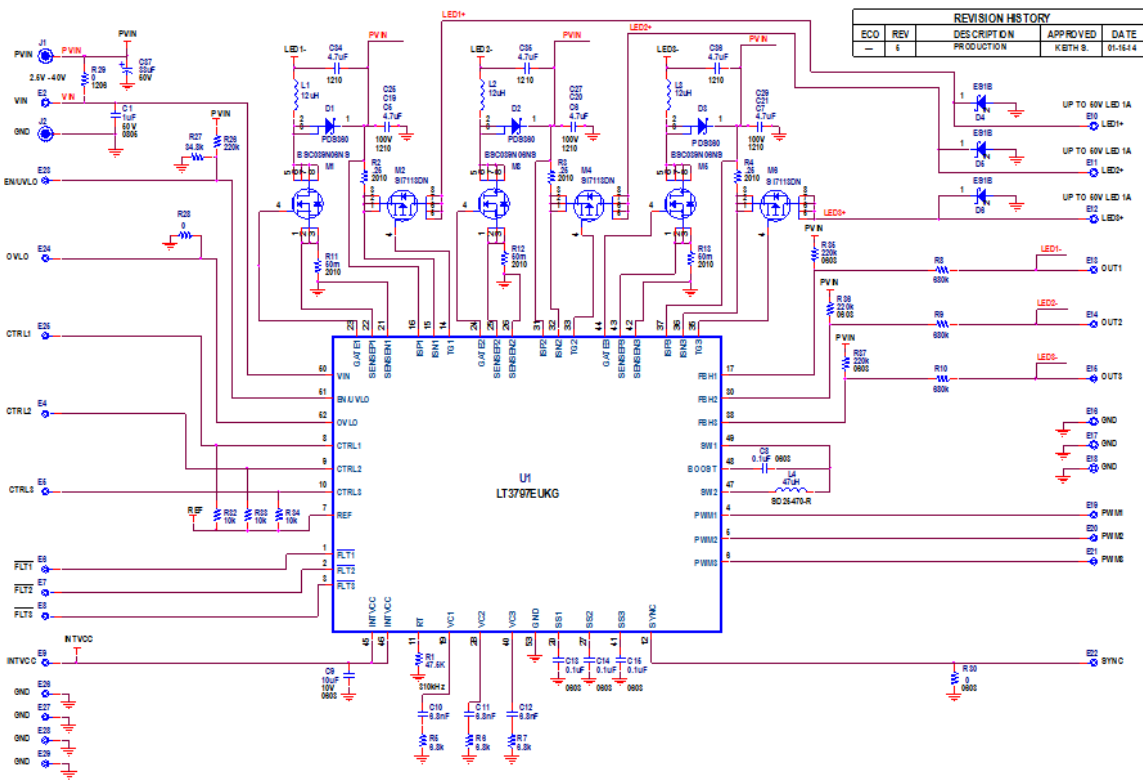
FIGURE 5-17: TOPOLOGY 1 FINAL DESIGN SCHEMATIC
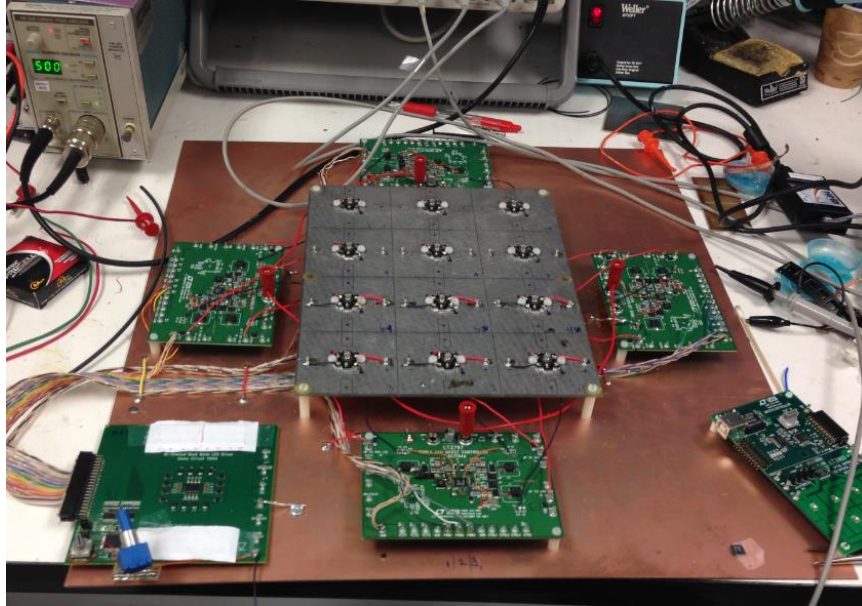
FIGURE 5-18: HARDWARE LAYOUT OF TOPOLOGY 1

FIGURE 5-19: FINAL HARDWARE LAYOUT OF TOPOLOGY 1

The final design schematic for topology 2 can be seen in figure 5-20. This includes all the final component values as well as the components listed in section 5.1. Figure 5-21 shows a picture of the final hardware layout.
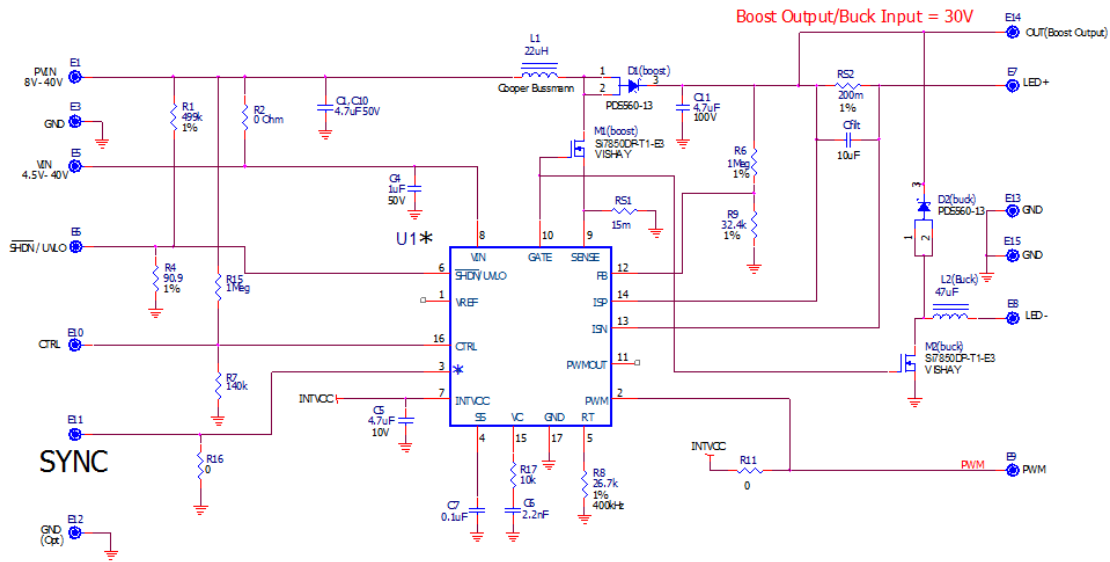
FIGURE 5-20: FINAL DESIGN SCHEMATIC OF TOPOLOGY 2
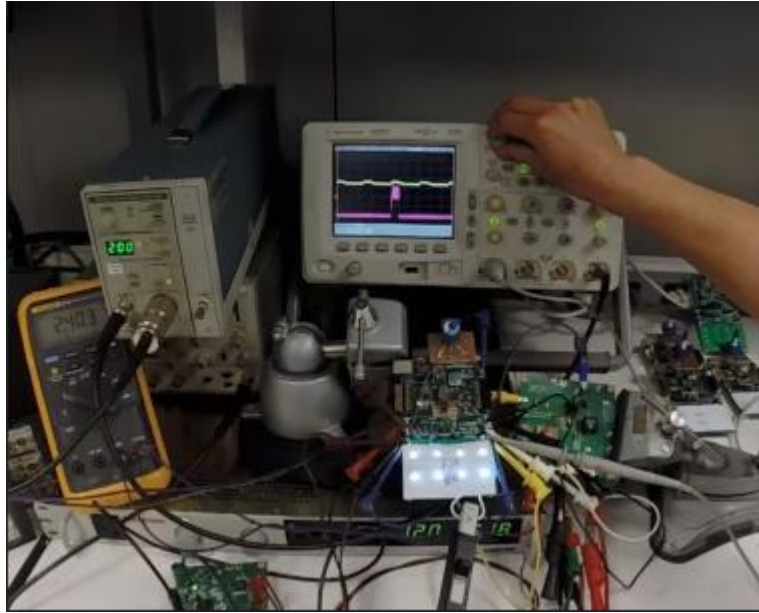
FIGURE 5-21: FINAL HARDWARE DESIGN FOR TOPOLOGY 2

The final design schematic for topology 3 can be seen in figure 5-22. This includes all the final component values as well as the components listed in section 5.1. Figure 5-23 shows a picture of the final hardware layout.
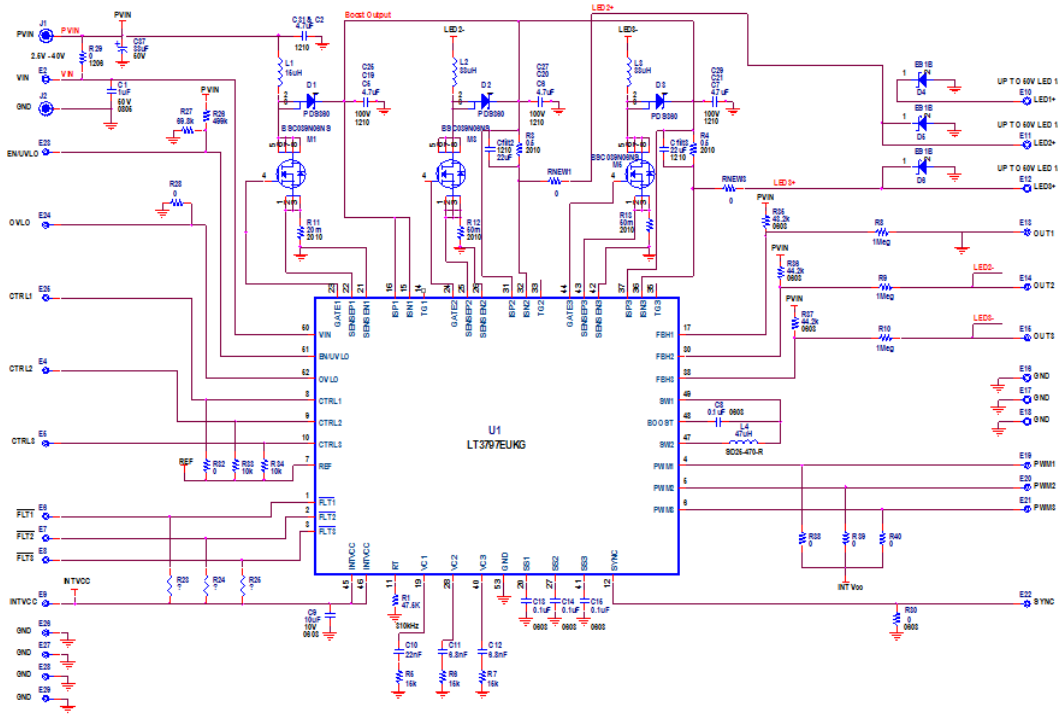
DC1784A

FIGURE 5-22: FINAL DESIGN SCHEMATIC FOR TOPOLOGY 3
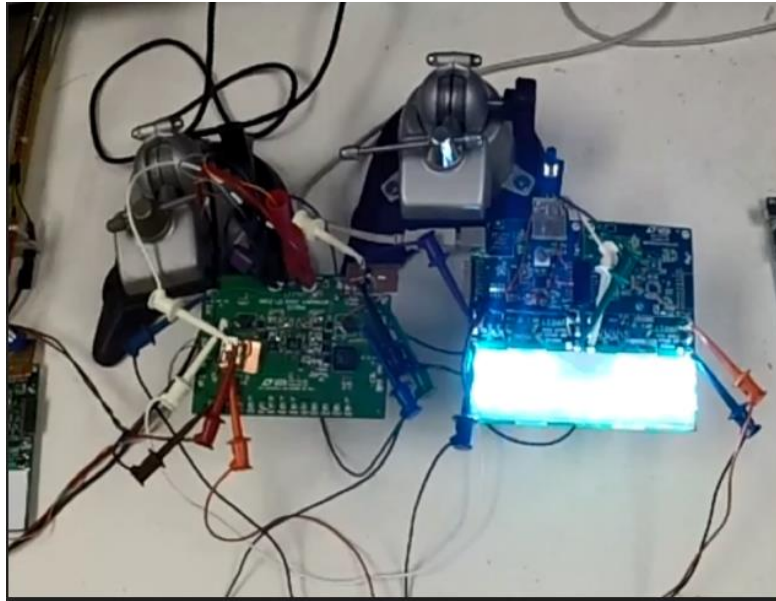
FIGURE 5-23: FINAL HARDWARE DESIGN FOR TOPOLOGY 3

Finally, the design of the PIC microcontroller can be seen in figure 5-25. The demo board, DC1199A, was used to integrate the PIC with the three topologies. This board was used because the PIC was already configured with the proper power components and USB interfacing needed to program the microcontroller. The only thing that was necessary to interface the DC1199A with the other boards was to create a ribbon cable that routed the signals from the output of the PIC to the PWM inputs.

FIGURE 5-24: FINAL DESIGN SCHEMATIC FOR PICLF877A ON DC1199A

With the circuits finalized and the code written for various patterns, this concludes the hardware verification for the adaptive lighting systems.

Chapter 6 : Conclusion

The objective of this thesis was to build and compare three different topologies using Linear Technology IC's that achieve adaptive lighting technologies. Table 6-1 shows the comparison of the three topologies based on the qualitative and quantitative data presented in chapter 5.

TABLE 6-1: COMPARISON OF HARDWARE RESULTS

	Topology 1	Topology 2	Topology 3
IC/Configuration	LT3797 Buck Mode	LT3755 Boost then Output Cap Free Buck Mode	LT3797 Dual Boost then Output Cap Free Buck Mode
Input Voltage	9-16V	9-16V	9-16V
Output Voltage	0-3V	0-24V	0-24V
Output Current	1A	500mA	500mA
PWM Dimming Ratio (@100Hz)	5000:1	500:1	500:1
Crossover Frequency [kHz]	38.88	5.1	9.8
Phase Margin [deg]	62.39	81.29	69.2
Gain Margin [dB]	10.41	11.27	11.17

Prior to the final designs of the three topologies, roadblocks were met that could have hindered the overall performance of the adaptive lighting systems. Based on the

load requirements, certain adjustments to the topology had to be made. The boost then buck topology was a new endeavor that had minimal testing done on it. After completing the tests shown in figure XXX, this boost then buck configuration is a viable option for the input and output characteristics of this application.

The removal of the output capacitor for loads configured in a string of LEDs resulted in a increase of crossover frequency of 140% (119Khz to 50kHz). While this new design feature allowed for higher PWM dimming ratios, the overall function of the topology did not change.

Due to the minimum on-time set by the IC's, there was speculation that adjustments to the switching frequency would have to be made. By increasing the switching frequency, the period would shorten, allowing the minimum on-time to be met. However, due to the on-resistance of the MOSFETs in the shorter outer, there was substantial voltage potential for the converters to maintain proper output current when all the LEDs were bypasses in the string.

Qualitative and quantitative data shows that topology 1 yields the highest PWM dimming ratio. Since each LED has its own converter, this higher priced, yet higher performance option may be viable for applications where LEDs need to be dimmed to a very low brightness. An application for this may involve airplane dash lights, where the ambient light of the cockpit is extremely low. Topologies 2 and 3 resulted in an equal PWM dimming ratio, which is expected, as they have a similar set up. This slower, lower cost alternatives would be useful in brake or headlight applications, where the dimming ratio only needs to go up to 100:1.

PWM dimming waveforms were created using C commands carried out by a PIC microcontroller. The outputs of the PIC were routed to the corresponding PWM pins of the IC. Since the patterns were implemented in software, cool patterns and designs could be tested for visual aesthetics.

Overall, this thesis successfully met the design requirements outlined in chapter 3.

REFERENCES

[1] Arik, M.; Petroski, J.; Weaver, S., "Thermal challenges in the future generation solid state lighting applications: light emitting diodes," Thermal and Thermomechanical Phenomena in Electronic Systems, 2002. ITHERM 2002. The Eighth Intersociety Conference on , vol., no., pp.113,120, 2002

[2] Chen, N. P. Handbook of Light Emitting and Schottky Diode Research. New York: Nova Science, 2009. Print.

[3] Cooper, Julian. "Driving LED Lamps – Some Simple Design Guidelines." Ledsmagazine.com. N.p., Aug. 2008. Web. 05 Feb. 2015.

[4] Falin, Jeff. "Compensating and Measuring the Control Loop of a High-power LED Driver." (n.d.): n. pag. Ti.com. Texas Instruments, Oct. 2008. Web. 4 Feb. 2015.

[5] Held, Gilbert. Introduction to Light Emitting Diode Technology and Applications. Boca Raton: CRC, 2009. Print.

[6] Kueck, Christian. "Power Supply Layout and EMI." AN139 (2012): n. pag. Linear.com. Linear Technology, Oct. 2012. Web. 4 Feb. 2015.

[7] Likely, Chris. "Guide to Selecting Inductors for Switching Regulators." Magnetic Components Special Report (2003): n. pag. Powerelectronics.com. Power Electronics Sponsor, July 2003. Web. 4 Feb. 2015.

[8] Linear Technology, "LT3755 - 40Vin,75Vout, LED Controllers," LT3755 Datasheet, 2013.

[9] Linear Technology, "LT3797 – Triple Output LED Driver Controller," LT3797 Datasheet, 2013.

[10] Microchip, "28/40/44-Pin Enhanced Flash Microcontroller," PIC16F877A Datasheet, November 2012.

[11] Perica, Goran. "Ceramic Input Capacitors Can Cause Overvoltage Transients." AN88 (n.d.): n. pag. Linear.com. Linear Technology, Mar. 2001. Web. 4 Feb. 2015.

[12] Richardson, Chris. "Driving LEDs: To Cap or Not to Cap." AN116 (n.d.): n. pag.Power.national.com. Texas Instruments, June 2007. Web. 4 Feb. 2015.

[13] Rosen, Rich. "Dimming Techniques for Switched-Mode LED Drivers." AN126 (2011): n. pag. Ti.com. Texas Instruments, June 2011. Web. 4 Feb. 2015.

[14] Seago, John. "AN76 - OPTI-LOOP Architecture Reduces Output Capacitance and Improves Transient Response." (1999): 1-16. Linear.com. 5 May 1999. Web. 4 Feb. 2015. [Online].

[15] Skovmand, Tim. "Micropower High Side MOSFET Drivers." (1993): n. pag. Linear.com. Linear Technology, Jan. 1993. Web. 4 Feb. 2015.

[16] Texas Instruments, ed. "Design Challenges of Switching LED Drivers." AN-1656 (2007): n. pag. Ti.com. Texas Instruments, May 2013. Web. 4 Feb. 2015.

[17] Williams, Jim. "AN25 - Switching Regulators for Poets: A Gentle Guide for the Trepidatious." Application Note 25 (1987): 1-24. Linear.com. Linear Technology, Sept. 1987. Web. 4 Feb. 2015.

[18] Williams, Jim, and Brian Huffman. "Some Thoughts on DC/DC Converters." (1988): n. pag. Linear.com. Linear Technology, Oct. 1988. Web. 4 Feb. 2015.

[19] Williams, Jim. "High Voltage, Low Noise, DC/DC Converters." AN118 (2008): n. pag. Linear.com. Linear Technology, Mar. 2008. Web. 4 Feb. 2015.

[20] Williams, Jim. "Step-Down Switching Regulators." (n.d.): n. pag. Linear.com. Linear Technology, Aug. 1989. Web. 4 Feb. 2015.

[21] Zhang, Henry J. "PCB Layout Considerations for Non-Isolated Switching Power Supplies." AN136 (2012): n. pag. Linear.com. Linear Technology, June 2012. Web. 4 Feb. 2015.

APPENDICES

Code:

```
#define ANALOG_SELECTOR

#include "DC590B.h"

#include "pic16f877sfrs.h"

#include "tables.h"

#use fast_io(A)

#use fast_io(B)

#use fast_io(D)

#use fast_io(E)

#use rs232(baud=9600,parity=N,xmit=PIN_B1,rcv=PIN_B2,bits=8)

#bit CLOCK_HI  = PORTB.0

#bit CLOCK_LO  = PORTB.1

#bit OE_       = PORTE.0

void all_off(void);

void all_on(void);

void steps_0p1(void);

void steps_1p0(void);

void full_range_lin(void);

void full_range_log(void);

void full_range_scroll(void);

void random_trippy(void);

void all_on(void);

void all_float(void);

void analog_in(void);

void read_analog(void); //function to read analog input

void gradual_increase(void);

void grad_increase_pulse(void);

void linear_day(void);
```

void linear_night(void);

void linear_day_200(void);

void linear_night_200(void);

void rotate_leds(void);

#inline

void one_cycle(void);

void one_cycle_lin(unsigned int x);

void one_cycle_lin_night(unsigned int x);

// 5500 delay is about 10ms cycle time

int16 CONST full_log_delay_table[16] = {1,2,3,5,8,13,22,36,60,101,169,282,471,786,1312,2192};

int16 CONST full_lin_delay_table[16] = {331,331,331,331,331,331,331,331,331,331,331,331,331,331,331,331};

int16 CONST lo_lin_delay_table[16] = {3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3};

int16 CONST test_table[128] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4, 5,5,5,5,5,5,6,6,6,6,7,7,7,7,8,8,8,9,9,9,10,10,10,11,11,12,12,13,13,14,14,15,15,16,16,17,1 8,18,19,20,20,21,22,23,24,25,26,27,28,29,30,31,32,33,35,36,37,39,40,42,44,45,47,49,51 ,53,55,57,59,61,64,66,68,71,74,77,80,83,86,89,93};

int16 CONST test_table_1[128] = {96,100,104,108,112,116,121,125,130,135,140,146,151,157,163,169,176,183,190,197,2 04,212,220,229,238,247,256,266,276,287,298,309,321,334,346,360,374,388,403,418,43 4,451,468,486,505,525,545,566,587,610,633,658,683,709,736,765,794,825,856,889,923 ,959,996,1034,1074,1115,1158,1202,1248,1296,1346,1398,1452,1507,1565,1625,1688, 1753,1820,1890,1962,2038,2116,2197,2282,2370,2461,2555,2653,2755,2861,2971,308 5,3204,3327,3455,3587,3725,3868,4017,4171,4332,4498,4671,4850,5037,5230,5431,54 50,5455,5460,5465,5470,5475,5480,5485,5485,5485,5485,5485,5485,5485,5485,5485,5 485,5485,5485,5485};

int16 CONST linear_table[128] = {417,433,449,465,481,497,513,529,546,562,578,594,611,627,644,660,676,693,709,726, 742,759,776,792,809,826,842,859,876,893,909,926,943,960,977,994,1011,1028,1045,1 062,1079,1097,1114,1131,1148,1165,1183,1200,1217,1235,1252,1270,1287,1305,1322, 1340,1357,1375,1393,1410,1428,1446,1464,1481,1499,1517,1535,1553,1571,1589,160 7,1625,1643,1661,1679,1697,1715,1733,1752,1770,1788,1807,1825,1843,1862,1880,18 99,1917,1936,1954,1973,1991,2010,2029,2047,2066,2085,2104,2122,2141,2160,2179,2

```c
198,2217,2236,2255,2274,2293,2312,2331,2350,2370,2389,2408,2427,2447,2466,2485,
2505,2524,2544,2563,2583,2602,2622,2641,2661,2680};


int16 CONST linear_table_1[128] =
{2700,2720,2740,2759,2779,2799,2819,2839,2859,2879,2899,2919,2939,2959,2979,299
9,3019,3039,3059,3080,3100,3120,3140,3161,3181,3202,3222,3242,3263,3283,3304,33
25,3345,3366,3387,3407,3428,3449,3469,3490,3511,3532,3553,3574,3595,3616,3637,3
658,3679,3700,3721,3742,3763,3785,3806,3827,3849,3870,3891,3913,3934,3955,3977,
3998,4020,4042,4063,4085,4106,4128,4150,4172,4193,4215,4237,4259,4281,4303,432
5,4346,4368,4391,4413,4435,4457,4479,4501,4523,4546,4568,4590,4612,4635,4657,46
80,4702,4724,4747,4769,4792,4815,4837,4860,4882,4905,4928,4951,4973,4996,5019,5
042,5065,5088,5111,5134,5157,5180,5203,5226,5249,5272,5295,5319,5342,5365,5388,
5412,5435};

//5640,5856,6081,6315,6558,6809,7071,7343,7625,7918,8222,8538,8866,9207,9560,99
28,10309,10705,11116};

int8 CONST lodata_table[16] =
{0xFE,0xFC,0xF8,0xF0,0xE0,0xC0,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};

int8 CONST hidata_table[16] =
{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFE,0xFC,0xF8,0xF0,0xE0,0xC0,0x80,0x00};

int8 CONST lodata_table_test[16] =
{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};

int8 CONST hidata_table_test[16] =
{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};

int8 lodata[16];

int8 hidata[16];

int16 delays[16];

int8 i;

unsigned int8 j;

int16 cycles;

void main()
  {
 // unsigned int x;
 char y;
  TRISA = 0xFF;
```

```c
    TRISB = 0;

    TRISC = 0;

    TRISD = 0x0F;//0

    TRISE = 0;



    setup_adc_ports(AN0);

setup_adc(ADC_CLOCK_INTERNAL);

initialize();


    OE_ = 0;//output_high(OE_);   //Turn off all lights

    while(1){

set_adc_channel(0);

    delay_us(20);

switch(input_A() & 0x1E)//3C

        {

        case 0x00: all_off();   break;          // Swich position 0

        case 0x10: linear_day_200(); break;        // Swich position 1

        case 0x02: linear_night_200(); break;      //switch position 2

        case 0x12: gradual_increase(); break;      // Swich position 3

        case 0x04: grad_increase_pulse();  break;    //Switch Position 4

        case 0x14: linear_day(); break;          // Swich position 5

        case 0x06: linear_night(); break;         // Swich position 6

        case 0x16: fade(); break;             // Swich position 7

        case 0x08: all_float(); break;         // Swich position 8

        case 0x18: thoutoone(); break;          // Swich position 9

        default: all_off(); break;

        }
```

```c
    }
  }// End of main()

void gradual_increase(void)
  {
   unsigned int x1;
   unsigned int16 temp;

    x1 = read_adc();
  if ( x1 >= 10 && x1 <= 56) //pedal pressed very lightly
  {
     for(i=0; i<16; ++i)
       {
         delays[i] = full_log_delay_table[i];
         lodata[i] = lodata_table[15];
         hidata[i] = hidata_table[15];
       }
         OE_ = 0;
     while((input_A() == 0x02) && ( x1 >= 10) && (x1 <= 56))
       {
         one_cycle();
         x1 = read_adc();
       }
  }
  else if ( x1 < 10) //pedal not pressed at all
  {
         OE_=0;
         output_C(0);
```

```c
        output_D(0);

        CLOCK_HI = 1;

        CLOCK_HI = 0;

    }

        else

    {

   if ( x1 <= 127)

        temp = (5500 - test_table[x1])/15;

      else


        temp = (5500 - test_table_1[x1-128])/15;

        for(i=0; i<16; ++i)

        {

            delays[i] = temp;

            lodata[i] = 0;

            hidata[i] = 0;

        }

          if (x1 <=127)

            delays[0] = test_table[x1];

          else

            delays[0] = test_table_1[x1-128];

          OE_ = 0;

           one_cycle();

    }

}

void grad_increase_pulse(void)

{

   unsigned int x1;
```

```
unsigned int16 temp;

  x1 = read_adc();

if ( x1 >= 10 && x1 <= 56) //pedal pressed very lightly

{

  for(i=0; i<16; ++i)

    {

      delays[i] = full_log_delay_table[i];

      lodata[i] = lodata_table[15];

      hidata[i] = hidata_table[15];

    }

      OE_ = 0;


  while((input_A() == 0x12) && ( x1 >= 10) && (x1 <= 56))

    {

      one_cycle();

      x1 = read_adc();

    }

}

else if ( x1 < 10) //pedal not pressed at all

{

      OE_=0;

      output_C(0);

      output_D(0);

      CLOCK_HI = 1;

      CLOCK_HI = 0;

}

else if (x1 >= 250) //pedal fully pressed

{
```

```c
      fade();
  }
  else
  {
    if ( x1 <= 127)
      temp = (5500 - test_table[x1])/15;
    else
      temp = (5500 - test_table_1[x1-128])/15;
    for(i=0; i<16; ++i)
      {
        delays[i] = temp;
        lodata[i] = 0;
        hidata[i] = 0;


      }
        if (x1 <=127)
        delays[0] = test_table[x1];
        else
        delays[0] = test_table_1[x1-128];
        OE_ = 0;
        one_cycle();
  }
}


void linear_day(void)
{
 unsigned int x;
 unsigned int16 temp;
```

```
x = read_adc();

  if (x <= 5)

    {

       OE_=0;

       output_C(0);

       output_D(0);

       CLOCK_HI = 1;

       CLOCK_HI = 0;

    }

  else if ( x > 5 && x <= 250) //linear increase of 12 LEDs

    {

       if (x <= 127)

         temp = (5500 - linear_table[x])/15;

       else

         temp = (5500 - linear_table_1[x-128])/15;

      for(i=0; i<16; ++i)

       {

         delays[i] = temp;

         lodata[i] = 0;

         hidata[i] = 0;

       }

         if (x <=127)

            delays[0] = linear_table[x];

         else

            delays[0] = linear_table_1[x-128];

         OE_ = 0;

         one_cycle_lin(x);

    }
```

```c
    else //full on
      {
       OE_=0;
       output_C(0xFF);
       output_D(0xF0);
       CLOCK_HI = 1;
       CLOCK_HI = 0;
      }
}
void linear_night(void)
{
 unsigned int x;
 unsigned int16 temp;
  x = read_adc();
    if (x <= 5)
      {
       for(i=0; i<16; ++i)

        {
        delays[i] = 30;
        lodata[i] = 0;//lodata_table[4];
        hidata[i] = 0;//hidata_table[4];
        }
       delays[0] = 2;
      OE_ = 0;//output_high(OE_);   //Turn off all lights
      while(input_A() == 0x06 && x<=5)
        {
        one_cycle();
```

```
      x = read_adc();

    }

  }

else if ( x > 5 && x <= 250) //linear increase of 12 LEDs

  {

    if ( x <= 127)

      temp = (5500 - linear_table[x])/15;

    else

      temp = (5500 - linear_table_1[x-128])/15;

   for(i=0; i<16; ++i)

    {

     delays[i] = temp;

     lodata[i] = 0;

     hidata[i] = 0;

    }

      if (x <=127)

        delays[0] = linear_table[x];

      else

        delays[0] = linear_table_1[x-128];

     OE_ = 0;

     one_cycle_lin(x);

     one_cycle_lin_night(x);

  }

else //full on

  {

  OE_=0;

  output_C(0xFF);

  output_D(0xF0);
```

```c
       CLOCK_HI = 1;

       CLOCK_HI = 0;

       }

}

/////////////////////////////////////////////////////////////////

void linear_day_200(void)

{

 unsigned int x;

 unsigned int16 temp;

  x = read_adc();

    if (x <= 5)

      {

        OE_=0;

        output_C(0);

        output_D(0);

        CLOCK_HI = 1;

        CLOCK_HI = 0;

       }

    else if ( x > 5 && x <= 250) //linear increase of 12 LEDs

      {

       if (x <= 127)

          temp = (2750 - (linear_table[x]/2))/15;

        else

          temp = (2750 - (linear_table_1[x-128]/2))/15;

       for(i=0; i<16; ++i)

        {

          delays[i] = temp;

          lodata[i] = 0;
```

```c
        hidata[i] = 0;

      }

        if (x <=127)

          delays[0] = linear_table[x]/2;

        else

          delays[0] = linear_table_1[x-128]/2;

        OE_ = 0;

        one_cycle_lin(x);

    }

  else //full on

    {

     OE_=0;

     output_C(0xFF);

     output_D(0xF0);

     CLOCK_HI = 1;

     CLOCK_HI = 0;

    }

}
void linear_night_200(void)

{

 unsigned int x;

 unsigned int16 temp;

  x = read_adc();

    if (x <= 5)

      {

       for(i=0; i<16; ++i)

        {

        delays[i] = 30;
```

```c
     lodata[i] = 0;//lodata_table[4];

     hidata[i] = 0;//hidata_table[4];

          //   lodata[i] = 0;

          //   hidata[i] = 0;

      }

    delays[0] = 2;

   OE_ = 0;//output_high(OE_);    //Turn off all lights

   while(input_A() == 0x02 && x<=5)

     {

     one_cycle();

      x = read_adc();

      }

   }

else if ( x > 5 && x <= 250) //linear increase of 12 LEDs

   {

     if (x <= 127)

        temp = (2750 - (linear_table[x]/2))/15;

      else

        temp = (2750 - (linear_table_1[x-128]/2))/15;

    for(i=0; i<16; ++i)

     {

      delays[i] = temp;

       lodata[i] = 0;

       hidata[i] = 0;

      }


        if (x <=127)

          delays[0] = linear_table[x]/2;
```

```c
        else

           delays[0] = linear_table_1[x-128]/2;

           OE_ = 0;

           one_cycle_lin(x);

           one_cycle_lin_night(x);

       }

     else //full on

       {

        OE_=0;

        output_C(0xFF);

        output_D(0xF0);

        CLOCK_HI = 1;

        CLOCK_HI = 0;

       }

}
//////////////////////////////////////////////////////////////////////
void analog_in(void)

  {

  unsigned int x;

  unsigned int y;

     OE_ = 0;

     x = read_adc();

     output_C(x);

     CLOCK_HI = 1;

     CLOCK_HI = 0;

   }
//all on 12 channel
```

```c
void all_on(void)
  {
  OE_=0;
  output_C(0xFF);
  output_D(0xF0);   //FF);
  CLOCK_HI = 1;//output_high(CLOCK_LO);
  CLOCK_HI = 0;//output_low(CLOCK_LO);
  while(input_A() == 0x16);  //14);
  }
//all float 12 channel

void all_float(void)
  {
  //OE_ = 1;
  while(input_A() == 0x08){ //15
    OE_ = 1;
  }
  OE_=0;
  }
//12 channel?
void full_range_scroll(void)
  {
  OE_ = 0;
  for(i=0;i<12;i++)        //i<16; ++i)
    {
    delays[i] = full_log_delay_table[i];
    lodata[i] = lodata_table[i];
    hidata[i] = hidata_table[i];
```

```c
        }
    while(input_A() == 0x12)//3)

        {
        while(--j)

            {
            one_cycle();

            }

        rotate_leds();

        j=25;

        }

    }
void rotate_leds(void)

    {
    int1 ms_bit0, ls_bit0;

    for(i=0; i<=11; ++i)

        {
        ls_bit0 = lodata[i] & 0x01;

        ms_bit0 = hidata[i] & 0x01;

        lodata[i] = lodata[i] >> 1;

        hidata[i] = hidata[i] >> 1;

        if(ls_bit0) hidata[i] |= 0x80;

        if(ms_bit0) lodata[i] |= 0x80;

        }

    }


void all_off(void)

    {
    OE_=0;
```

```c
    output_C(0);

    output_D(0);

    CLOCK_HI = 1;

    CLOCK_HI = 0;

    while(input_A() == 0)

      {  // wait for change

      output_C(0);

      output_D(0);

      CLOCK_HI = 1;

      CLOCK_HI = 0;

      }

  }

//12 channel

void full_range_lin(void)

  {

  for(i=0; i<12; ++i)

    {

    delays[i] = full_lin_delay_table[i];

    lodata[i] = lodata_table[i];

    hidata[i] = hidata_table[i];

    }

  OE_ = 0;//output_high(OE_);   //Turn off all lights

  while(input_A() == 0x10)//1)

    {

    one_cycle();

    }

  }

//12 channel
```

```c
void full_range_log(void)
  {
  for(i=0; i<16; ++i)
    {
    delays[i] = full_log_delay_table[i];

    lodata[i] = lodata_table[i];

    hidata[i] = hidata_table[i];

    }
  OE_ = 0;//output_high(OE_);   //Turn off all lights

  while(input_A() == 0x12)
    {
    one_cycle();

    }
  }


//one cycle 12 channel
#inline
void one_cycle(void)
  {
  int i = 0;
  output_C(0xFF);

  output_D(0xF0);          //FF);

  CLOCK_HI = 1; CLOCK_HI = 0;

  for(i=0; i<16; ++i)
    {
    cycles = delays[i];

    while(--cycles);

    output_C(hidata[i]);
```

```c
        output_D(lodata[i]);

        CLOCK_HI = 1; CLOCK_HI = 0;

        }

    }

void one_cycle_lin(unsigned int x)

{

    int i = 0;


        if(x <= 42)//6

        {

            output_C(0xFC);

            output_D(0x00);

        }

        else if (x > 42 && x<=84)//7

        {

            output_C(0xFE);

            output_D(0x00);

        }

        else if (x > 84 && x<=126)//8

        {

            output_C(0xFF);

            output_D(0x00);

        }

        else if (x > 126 && x<=168)//9

        {

            output_C(0xFF);

            output_D(0x80);

        }
```

```c
        else if (x > 168 && x<=210)//10

        {

            output_C(0xFF);

            output_D(0xC0);

        }

        else if (x > 210 && x<=250)//11

        {

            output_C(0xFF);

            output_D(0xE0);

        }

        else

        {

            output_C(0xFF);

            output_D(0xF0);

        }

    CLOCK_HI = 1; CLOCK_HI = 0;

    for(i=0; i<16; ++i)

        {

        cycles = delays[i];

        while(--cycles);

        output_C(hidata[i]);

        output_D(lodata[i]);

        CLOCK_HI = 1; CLOCK_HI = 0;

        }

    }

void one_cycle_lin_night(unsigned int x)

{

    int i = 0;
```

```c
if(x <= 42)//6
{
  output_C(0x03);
  output_D(0xF0);
}
else if (x > 42 && x<=84)//7
{
 output_C(0x01);
  output_D(0xF0);
}
else if (x > 84 && x<=126)//8
{
  output_C(0);
  output_D(0xF0);
}
else if (x > 126 && x<=168)//9
{
  output_C(0);
  output_D(0x70);
}
else if (x > 168 && x<=210)//10
{
  output_C(0);
  output_D(0x30);
}
else if (x > 210 && x<=250)//11
{
  output_C(0);
```

```
        output_D(0x10);

    }

    else

    {

      output_C(0);

      output_D(0x00);

    }

  CLOCK_HI = 1; CLOCK_HI = 0;

  for(i=0; i<16; ++i)

    {

      delays[i] = 30;

      lodata[i] = 0;

      hidata[i] = 0;

    }

      delays[0] = 2;

      OE_ = 0;

      one_cycle();

}


  void one_cycle2(void)

  {

  int i = 0;

  output_C(0xFF);

  output_D(0x00);

  CLOCK_HI = 1; CLOCK_HI = 0;

  for(i=0; i<2; ++i)

    {

    cycles = delays[i];
```

```c
        if(cycles){
            while(--cycles);
        }
        output_C(hidata[i]);
        output_D(lodata[i]);
        CLOCK_HI = 1; CLOCK_HI = 0;
        }
    }
//12 channel
 void thoutoone(void){
   for(i=0; i<16; ++i)
      {
      delays[i] = full_log_delay_table[i];
      lodata[i] = lodata_table[15];  //15
      hidata[i] = hidata_table[15];  //15
      }
   OE_ = 0;//output_high(OE_);    //Turn off all lights
   while(input_A() == 4)
      {
      one_cycle();
      }
   }
//12 channel
 void huntoone(void){
   for(i=0; i<16; ++i)
      {
      delays[i] = 30;
      lodata[i] = 0;//lodata_table[4];
```

```
        hidata[i] = 0;//hidata_table[4];
   //    lodata[i] = 0;
   //    hidata[i] = 0;
        }
     delays[0] = 2;
   OE_ = 0;//output_high(OE_);    //Turn off all lights
   while(input_A() == 0x14)
      {
      one_cycle();
     }
     }
//12 channel
   void fade(void){
      int16 fadecount;
      int16 invf;
      int dir= 1;
      int temp = 2;
      int i =0;
      unsigned int x1;
      fadecount =500;
     x1 = read_adc();
      for(i=0;i<16;++i){
        delays[i] = 0;
        lodata[i] = 0xFF;
        hidata[i] = 0;
       }
      while(input_A()==6 || ((input_A() == 0x04) && (x1 >= 250))){
      x1 = read_adc();
```

```c
    if(fadecount>=500){

      dir=1;

      fadecount=500;

    }

    if(fadecount<=1){

      dir = 0;

      fadecount=0;

    }

    if(dir==1){

      fadecount=(fadecount-5);

    }

    else{

      fadecount=(fadecount+5);

    }

    invf=500-fadecount;

    delays[0] = fadecount;

    delays[1] = invf;

      //lodata[0] = 0xFF;

      //hidata[0]= 0xFF;

    while(--temp){

      one_cycle2();

    }

      temp = 2;

    }

  }
///////////////////////////////////////////////////////////////////
// Hardware Initialization
///////////////////////////////////////////////////////////////////
```

```c
void initialize(void)

  {
  output_high(ISO_PWR_SD_);  //turn on power

  setup_psp(PSP_DISABLED);

  setup_counters(RTCC_INTERNAL,RTCC_DIV_1);

  setup_timer_1(T1_DISABLED);

  setup_timer_2(T2_DIV_BY_1,0,1);
//  setup_timer_2(T2_DISABLED,0,1);

  setup_comparator(NC_NC_NC_NC);

  setup_vref(FALSE);

  output_low(PIN_E0);


// I/O Initialization
  input(RXF_);

  input(TXE_);

  output_high(RD_);

  output_low(WR);

  }
```