ABSTRACT

PINDICURA, THARUNACHALAM. Analysis of Microcontroller Based High Brightness LED Drivers—a Cost Oriented Approach to Understand the Relation between Computational Requirements and DC-DC Converter Components. (Under the direction of Dr. Alexander Dean).

High Brightness LEDs are replacing traditional incandescent and fluorescent light bulbs in several lighting applications. HB LEDs are known to be more energy efficient and more reliable than conventional lighting technologies. Several microcontroller based driver topologies have been developed to power HB LEDs with focus on energy efficiency and cost. These drivers also implement several vital features like dimming and over-temperature protection. But these studies have not explored the relation between the microcontroller's computational requirements and the size of active components of the driver. In this study we focus on reducing the overall cost of a three channel HB LED driver by optimizing the software and hardware requirements. The results show that the overall cost of the HB LED driver can be minimized significantly for given specifications by choosing the optimum combination of active components and microcontroller.

Analysis of Microcontroller Based High Brightness LED Drivers—a Cost Oriented Approach to Understand the Relation between Computational Requirements and DC-DC Converter Components

by
Tharunachalam Pindicura

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Computer Engineering

Raleigh, North Carolina

2012

APPROVED BY:

Dr. Alexander Dean Committee Chair	Dr. Subhashish Bhattacharya
Dr. James Tuck	

DEDICATION

To my parents

BIOGRAPHY

Tharunachalam Pindicura was born on February 2, 1989 in the city of Hyderabad, Andhra Pradesh, India. He did his primary and secondary schooling in his hometown of Hyderabad. He pursued undergraduate studies at Indian Institute of Technology – Kharagpur, West Bengal, India and received his bachelor’s degree (Bachelor of Technology) in Electrical Engineering, in July 2010. He worked as an intern at Embedded Systems Laboratory at Central Mechanical Engineering Research Institute, Durgapur, India during the summer of 2009. After the completion of his undergraduate studies he joined as a graduate student at the Department of Electrical and Computer Engineering, NCSU. He is majoring in computer engineering and his areas of interest are computer architecture and embedded systems. After graduating from NCSU as a Master of Science in computer engineering he will be joining IBM Austin as a hardware engineer.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Alexander Dean for the opportunity to work under him and for his invaluable guidance throughout the course of this project. I have gained valuable research experience while working with him. I would also like to thank Dr. Subhashish Bhattacharya for his immense support and guidance. I am thankful to Dr. James Tuck for being a part of my advisory committee and for his valuable feedback. My fellow group members Avik Juneja, Mihir Shah and Michael Plautz have always been very helpful. To all my friends who have made my stay at NCSU so memorable, thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1 Introduction

1.1 Significance of the study

High Brightness LEDs (HB-LED) are more energy efficient than conventional illumination techniques like incandescent lights. Until recently the HB-LEDs could not compete with traditional lighting methods due to their higher cost. But latest developments have made HB-LEDs cost effective. These developments make HB-LEDs an energy efficient, more reliable and cost effective alternative to incandescent illumination. HB-LEDs are already the preferred lighting option in automobiles, monitor backlighting, mobile devices and signage. They have been successfully implemented in automobile headlights.

LED drivers play an important role in improving the energy efficiency and lifetime of the HB-LED. HB-LED requires a constant current flowing through it for maximum efficiency. DC-DC converters with closed loop control are employed to ensure the constant current flow. The LED driver also implements over - current protections and over temperature protection to increase the lifetime of the HB-LED. LED driver also allows dimming of the LED to help in energy saving. In order to implement these features a microcontroller is used. The microcontroller runs tasks which implement the control loop, dimming and protections. A single microcontroller handles the workload of several LED strings. Hence the microcontroller must be able to effectively schedule many tasks to achieve optimum performance for the HB-LED module.

1.2 Motivation

Available studies discuss the essential features of HB-LED drivers and different topologies to implement them. But they do not explore the timing constraints on the scheduler which implements these features and the impact of microcontroller on the cost of the HBLED driver. This research work focuses on finding optimal selection of microcontroller, capacitor and inductor that would minimize the cost of the HBLED driver.

The selection of capacitor, inductor and microcontroller are all dependent on the specifications of the HB LED driver. Choosing a particular value for a component will put some constraints on another component. The cost of the inductor, capacitor and microcontroller are related. Hence to reduce the overall cost of the microcontroller we need to find an optimum point in design space. This point will indicate the type of microcontroller that is most preferred for designing HB LED drivers.

1.3 Preview of the work

A HB-LED driver is designed in order to evaluate its performance. The HB-LED driver is built based on boost topology DC-DC converter and is supported by a RL78 microcontroller. Boost topology is chosen as it is a common technique used in battery run LED applications. In these applications the voltage needed to drive the required current through the HB-LED is higher than the battery voltage and hence boost topology is preferred. This HB-LED driver has over-current protection, over-temperature protection and dimming. The micro-controller controls the output of the DC-DC converter by varying the duty cycle of two signals: boost gate and dimming gate. The basic outline of the module is given in Figure 1-1.

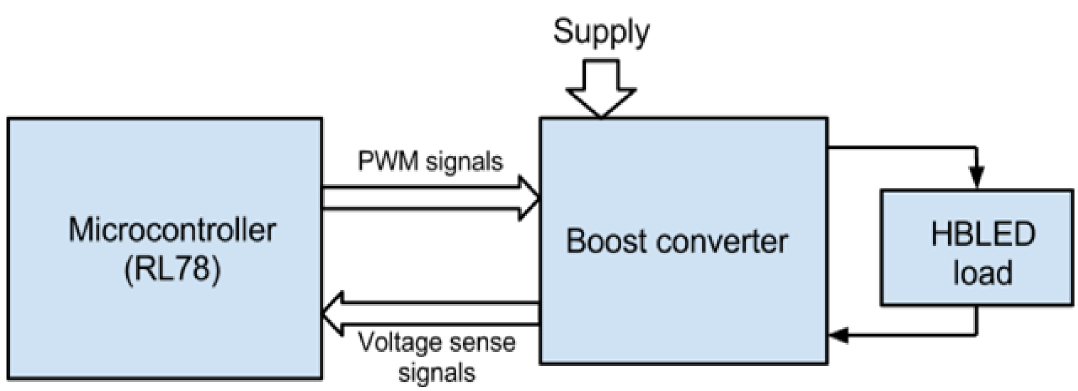

Figure 1-1. Schematic of the HB-LED driver

The boost converter takes the input from a power supply and boosts the voltage to the level required to drive the HB-LED load. The HB-LED load could be a single HB-LED or a string of HB-LEDs. The ideal load currents for HB-LEDs are in the range 300 mA to 3 A. The load current is measured and sent as a feedback to the microcontroller. The microcontroller implements a closed loop control by setting the duty cycle of the boost converter based on the current feedback and the required current. Thus the closed loop control maintains a constant load current. The microcontroller also keeps track of the current through the boost inductor. If there is an overcurrent condition the microcontroller reacts to prevent damage to the inductor. This overcurrent protection is vital to ensure a long lifetime for the HB-LED driver. The temperature of the HB-LED junction is also monitored by the microcontroller. High junction temperature lowers the luminous efficiency of the HB-LED. Extreme temperature can also cause permanent damage to the HB-LED lowering its lifetime. The microcontroller takes steps to prevent over-temperature conditions. Another feature implemented by the microcontroller is dimming. Dimming is an important feature that saves energy when the luminous output required is low. Dimming also helps in implementing over-temperature protection by lowering the power dissipated at the HB-LED junction. Dimming is incorporated in the design with a dimming transistor in series with the load. Rapid switching of this transistor results in a dimming effect of the HB-LED. This method is employed instead of varying the load current. Varying the load affects the chromaticity of the HB-LED. Rapid switching of the load current maintains the constant ON current but changes the average load current.

1.4 Outline for rest of the document

The rest of the document is organized as follows. Chapter 2 discusses some of the available HB-LED drivers and their features. The design and implementation of the boost topology based HB-LED driver is covered in Chapter 3 and Chapter 4 respectively. Chapter 5 is for the results obtained and analysis of these results. Chapter 6 contains the conclusions drawn form of this research and the prospects for future work.

CHAPTER 2

2 Related Work

In this chapter we will look into some of the existing high brightness LED solutions. Several companies are currently marketing either their HB LED drivers or micro-controllers that are designed for HB LED applications. The focus of their work is on energy efficiency, reliability, cost, and ease of development. Some of the devices and their features are discussed below.

2.1 Freescale MC9S0MP16 microcontroller [1]

In the application note AN4105 from Freescale, the authors describe an HB LED design based on MC9S0MP16 microcontroller. This is an 8 bit, 40 MHz microcontroller and is selected to develop a low-cost HB LED driver. The SEPIC topology DC-DC converter is used in the design. Some of the important features of this design are as follows.

1 Maximum output current of 0.5 A with a ripple of 10%

2 Switching frequency for the DC-DC converter gates is 350 kHz

3 PWM dimming at 100 Hz with a resolution of 114 steps

4 Closed loop PID control

The control loop takes the current through HB LEDs as the feedback. The ADC samples collected for feedback to the control are timed to synchronize with the PWM dimming. The microcontroller waits for a fixed delay after the dimming is high before starting to collect ADC samples. The PID control latency is 10 ms (100 Hz) or lower. The design implements several protections to improve the performance and reliability of the driver. The design can detect open load or short circuit conditions and take precautionary measures. There are protections against output over voltage, supply overvoltage and supply overcurrent. Over Temperature protections is also implemented and the HB LEDs are dimmed to 50% whenever the over temperature protection is triggered.

2.2 Texas instruments C2000 MCUs [2]

A white paper from Texas Instruments describes two designs of HB LED driver using C2000 microcontrollers from TI. These are 32 bit, 60 MHz microcontrollers. First design is designed with SEPIC topology based single channel DC-DC converter. A single channel drives 8 LED strings and each string has individual dimming control. The dimming is achieved by changing the duty cycle of the PWM signal. In this approach cost of the driver is reduced by having only one channel. But this prevents control of current flowing through individual HB LED strings. Each HB LED string requires a different voltage to operate to optimum load current as individual strings can have different number of LEDs. Even if the all strings have same number of LEDs, they have different voltage requirements due to process variations. Hence this design trades efficiency for cost.

The second design has two SEPIC and six boost channels. Each HB LED string is driven by a separate channel. The dimming is implemented by controlling the current flow through the LED strings. In this approach the luminous efficiency is compromised as the LED are forced to operate at sub-optimal currents. The chromaticity of the LEDs will also be affected when they are dimmed by varying the current.

Both the designs have temperature sensing for over temperature protection. Other protections can also be implemented for higher reliability. Energy saving by dimming the LEDs based on ambient light is also proposed.

2.3 Maxim MAX16826 [3]

This is programmable HB LED driver from Maxim. It is controlled by an external microcontroller. It has four channels and can be configured to operate as SEPIC or boost topology. Dimming is implemented with 45 Hz PWM signal. The switching frequency for the gates of DC-DC converters can be set between 100 kHz and 1 MHz. High switching frequency allows use of smaller inductors. This reduces the cost of the driver. The driver communicates with an external microcontroller with an I2C interface. The driver has LED string short, overvoltage and over temperature protections. There a six modes of operation for the driver. These modes are for normal operation and the different protection schemes.

2.4 Microchip PIC12HV615 [4]

Microchip document AN1138 describes implementing a HB LED driver with PIC12HV615 microcontroller. This is a low cost HB LED driver. PIC12HV615 is 8 MHz microcontroller. The design based on a single channel buck converter. It delivers 350 mA current with a ripple of 40%. The closed loop PI control is used to maintain constant current through the LED string. The control loop has a latency of 1.025 ms (976 Hz). The gate of the buck converter is switched at 125 kHz. The duty cycle of the switching frequency can be changed with a resolution of 64 steps. The document suggests use of a temperature sensor for over temperature protection. This design does not have a dimming control. This approach gives more significance to the lowering the cost of the driver and compromises on utility, energy efficiency and performance.

In the designs discussed above the goals have been energy efficiency, reliability, performance, cost and ease of use. In order to achieve these goals the drivers have various features like dimming and several protections. These features are implemented with the aid of a microcontroller. Hence the microcontroller has to run several time bound tasks for the efficient and reliable operation of the driver. The emphasis in these designs have been on choosing design topology and features that would result in achieving desired balance of energy efficiency, reliability, performance, cost and ease of use.

CHAPTER 3

3 Design of HB LED driver

3.1 Hardware Design

DC-DC driver are preferred over linear regulator in high brightness LED applications. Linear regulators are cost effective when the LED load requires low current. They are energy inefficient for high current LED loads. Hence DC-DC switching converters are used in HB LED applications. The standard topologies of DC-DC converters are:

1. Buck converter - Output voltage is lower than input voltage.
2. Boost converter - Output voltage is higher than input voltage.
3. SEPIC (Single Ended Primary Inductor Converter) - Output voltage can be lower or higher than input voltage.

3.1.1 Boost Converter

HB LEDs have a forward bias voltage in the range of 5V to 12V. Input voltage depends on the application. In mobile applications like handheld flashlights the battery voltage is lower than required. Boost topology is suited for applications where input battery voltage is lower than the forward bias of the HB LED. The cost analysis of boost topology HB LED driver can be extended to other topologies as the components of boost converter are similar to those used for other topologies. Since the focus is on the cost analysis of HB LED driver, the topology choice is not critical. Boost topology is selected for evaluating the real time embedded aspects of HB LED drivers.

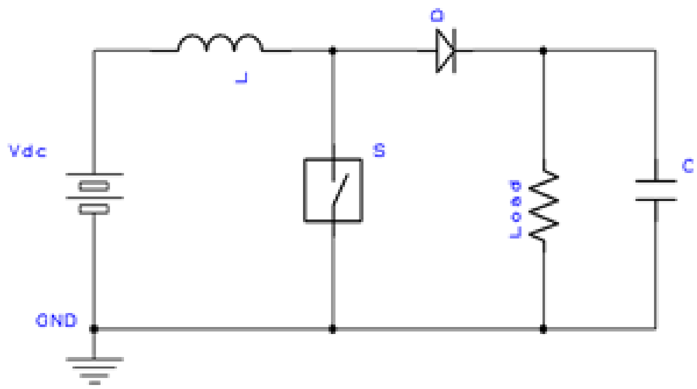

Figure 3-1. Boost converter topology

The general boost converter topology is shown in Figure 3-1. It has an inductor, a switching transistor, a diode and a capacitor. The switching transistor is continuously toggled using a PWM wave. When the switching transistor is closed the inductor current increases and energy is stored in the inductor increases. Opening the switch makes the inductor current charge the capacitor through the diode. The output voltage is maintained constant by the capacitor. The output voltage is determined by the duty cycle of the switching signal.

$$V_{out} = \frac{1}{1-D} \times V_{in}$$

Illumination applications often require three channel HB LED drivers, one each for red, green and blue HB LED strings (RGB). A three channel HB LED driver is designed to reflect the real time embedded requirements of such illumination applications. Each of the three channels is driven by a separate boost converter and has the following features:

1. Closed loop PID control for load current
2. PWM dimming
3. Over temperature protection
4. Overcurrent protection

3.1.2 Closed loop control

Closed loop control is essential to maintain the constant current required by the HB LED load. The load current is measured using a current sensing resistor. The voltage drop across resistor in series with HB LED load is measured using an ADC channel of the microcontroller. This feedback is used by the microcontroller to compute the response based on the control equations. Since the current requirement of a HB LED load is constant the control loop can have low latency.

Proportional - Integral - Derivative control is a common technique used for closed loop control. The feedback based on digital PID control is given by the following equations:

$$D_i = K_p \times e_i + K_i \times s_i + K_d \times (e_i - e_{i-1})$$

$$s_i = s_{i-1} + e_i$$

3.1.3 PWM dimming

Dimming is an important feature for HB LEDs. It provides a control for the illumination of the HB LED. HB LED can be dimming in low light conditions in order to save power. Dimming reduces the power dissipated at the HB LED junction and hence can be used to enforce temperature protection. Dimming is implemented by lowering the average current through HB LED. This can be done by lowering the output voltage of the driver. In this approach the HB LED operates at suboptimal current. This reduces the energy efficiency of the HB LED. Lowering the current level also affects the wavelength of the light output. Thus this approach is avoided in most commercial HB LED Drivers. The second approach is to use pulse width modulated signaling to rapidly switch current through the HB LED load. This is done by having a switching transistor in series with the HB LED load. In this approach the HB LED is turned on and off at a high frequency. This reduces the light output and results in dimming. The average current flowing through the HB LED load is altered without affects the current flowing through it during on time. The switching frequency should be high enough that there is no visible flicker in LED output. Human eye cannot notice any flicker at switching frequencies above 60 Hz. The duty cycle of the PWM signal determines the level of dimming.

There is one hazard when dimming is implemented with PWM signaling. When the load current path is cut off by the dimming transistor there is a rise in the output voltage. This spike is caused by the inductor dumping its energy on the load capacitor as there is no other current path. This phenomenon is explained in the document from Freescale which was discussed in chapter 2. (AN4105). The solution suggested to avoid this hazard was to stop the switching of the boost transistor when the dimming transistor is off.

PWM dimming requires two modifications to the regular functioning of control loop. Firstly the control loop should not collect any samples of load current when the dimming transistor is off. The load current requires a small time period to recover once the dimming transistor is switched on. The sampling of load current can begin only after the recovery. Synchronized sampling is implemented by having a fixed wait period after the dimming transistor turns on, before the load current sampling begins. Second modification required is that the control loop should not alter its output during the dimming off period. Hence the control loop task can be suspended between the switching off of dimming transistor and the recovery of load current after the dimming transistor is turned on.

3.1.4 Choosing the components

The HB LED loads used in experimentation are HB LED strings. These strings are rated 1A current. 12 V input is required to drive this current through the HB LED string.

Following are the design specifications:

Output voltage: 12V for 50 % Duty cycle.

Switching frequency: 100kHz

Load current: 1A

Current ripple: 30%

3.1.4.1 Inductor

The inductance required for the boost converter is given by the following equation.

$$L = \frac{V_{in} \times (V_{out} - V_{in})}{f_{SW} \times V_{out} \times \Delta I}$$

L = Inductance

V_{in} = Input voltage

V_{out} = Output voltage

f_{SW} = Switching frequency

ΔI = Current ripple

For 6V input, 12V output, ripple 300mA current and switching frequency of 100 kHz The inductor should be 100 µH.

3.1.5 Over-temperature protection

Over temperature protection is essential for reliability and energy efficiency of the HB LED. The light out of the HB LED degrades exponentially with time [5]. The degradation is given by:

$$L = L_o \times e^{-\alpha t}$$

L = Light output after time t

L_o = Initial light output

α = Light output degradation rate

t = Time

The light output degradation rate α depends on the junction temperature. Higher junction temperature leads to higher light output degradation rate. This implies the HB LED loses its light output efficiency quicker. The lifetime of the HB LED is also reduced. Hence restricting the junction temperature to acceptable levels is crucial to the lifetime of the HB LED.

It is not practical to monitor the junction temperature directly. The temperature of the junction is estimated by the temperature of the heat sink or a point that is specifically designed for this by the manufacturer. The temperature of the junction depends on the ambient temperature and the power dissipation. The power dissipated is controlled by controlling the average current through the HB LED. Dimming provides an effective method to alter the average load current. When the temperature recorded approaches a threshold dimming is activated to reduce the power output and lower the junction temperature.

Another aspect of over-temperature protection is the complete damage of the HB LED when the junction temperature exceeds a rated temperature level. Dimming may not be sufficient to prevent this condition. When such an HB LED is detected approaching such a extreme temperature the driver needs to be switched off and restarted only after the temperature falls below an acceptable level.

3.1.6 Over current protection

Over current protection is essential to protect the boost inductor. The inductor current when the boost transistor is ON should not exceed the rated current. A low ohm resistor is connected in series with the boost transistor. The voltage drop across this resistor is recorded by an ADC channel of the RL78 microcontroller. When the current approaches the maximum rated current of the inductor (1A) the boost gate is immediately switched off for the rest of the switching cycle. It is switched on only after ensuring that the load current is being drawn from the driver. This is detected by the measuring the voltage drop across the resistor used to record the load current.

3.2 Software Design

The software is comprised of running several tasks by a fixed priority, run to completion scheduler. The tasks that are required for the operation of the HB LED driver are as following.

1. Setting duty cycle ‘on time’.
2. Control task.
3. Dimming.
4. Over current protection.
5. Over temperature protection.

The frequencies at which these tasks run are spread over a wide range. For PWM operation in rl78 microcontroller the ‘on time’ cycles for the PWM signals are set at the end every boost signal cycle. Hence the frequency of this task is the frequency of the boost signal. This task has the highest frequency and requires the lowest computation cycles. This task also has the highest priority. If this tasks misses or is delayed it directly alters the duty cycle of the boost signal and the output voltage of the driver is affected.

Control task implements the closed loop control. Its task frequency is the control loop frequency. The maximum control loop frequency is limited by the boost signal frequency. In [] it is shown that the control loop frequency should be at least 6 times the boost signal frequency. This task requires the highest number of compute cycles. This task has ADC conversion for the current feedback and the control equations of the PID controller. The frequency of this task can be set by the user. The frequency of this task determines the response time of the closed loop control.

Dimming task is used to implement the PWM dimming for the HB LED load. The maximum frequency of this task is 60 Hz. This task is sets the 'on time' for the dimming signal. This is task has low priority. A delay in this task does not adversely affect the output. If the delay is larger than a single step of dimming, a change in the luminous output of the HB LED will be noticed.

Over current protection task implements the over current protection for the inductor. This task measures the current flowing through the inductor and suspends the boost converter operation if the value is within 10% of the current rating of the inductor. This task has high priority as its execution is critical to avoid permanent damage to the LED driver. The frequency of this task is determined by the open loop response time of the HB LED driver.

Over temperature protection task implements the over temperature protection for the HB LED junction. This task measures the junction temperature of the HB LED and keeps it from crossing a preset limit. This task lowers the dimming signal duty cycle when the temperature approaches the preset limit for temperature. This reduces the power dissipation at the HB LED junction and reduces its temperature. There is secondary temperature limit and exceeding this limit will cause permanent damage to the HB LED load. This task also checks if the junction temperature is approaching this secondary limit and prevents damage by switching off the boost converter driving the HB LED load. This task has a mid-level priority. The temperature rise at the junction depends on the power dissipation and the ambient temperature. As the power dissipation and the ambient temperature do not vary in short time duration the frequency of this tasks is low. Recurrent delay in this task makes the HB LED to operate at higher than optimum temperatures. This reduces the lifetime of the HB LED load.

3.3 Dimming

Task jitter in dimming affects the average current output of the driver. Figure 3-2shows the waveform for a dimming signal of time period $T_{dimming}$with an dimming 'on' time of T_{dim-on}. The dimming task has a task jitter of ΔT. The waveform for resulting output current with a peak current of $I_{pk}$is shown in Figure 3-3.

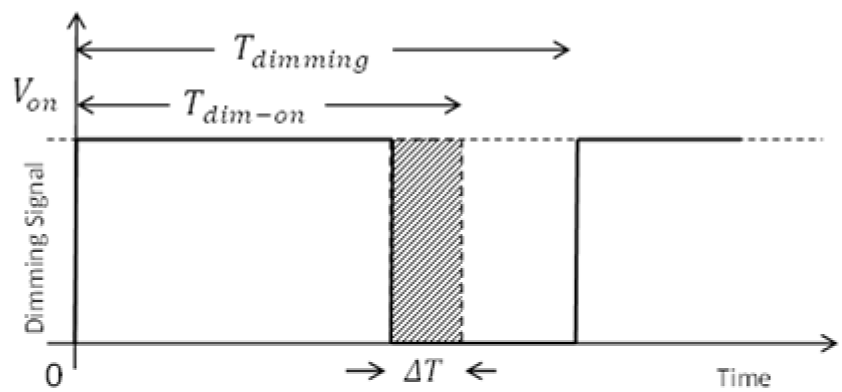

Figure 3-2. Dimming signal with task jitter in dimming task

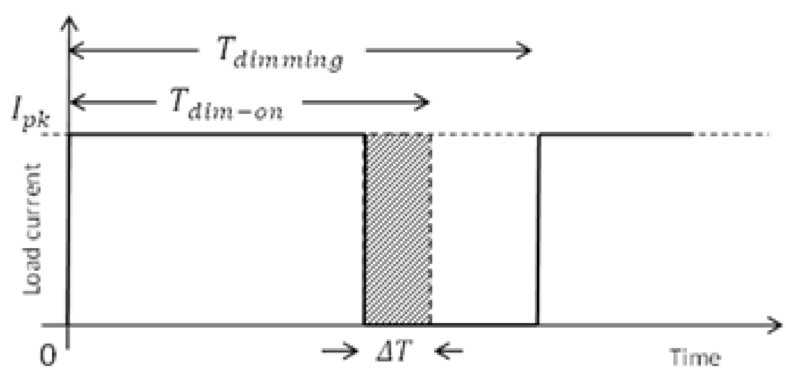

Figure 3-3. Output current with jitter in dimming task

Average output current without task jitter is $I_{avg} = \frac{T_{dim-on} \times I_{pk}}{T_{dimming}}$

Average output current with task jitter is $I'_{avg} = \frac{(T_{dim-on} - \Delta T) \times I_{pk}}{T_{dimming}}$

Change in output current due to dimming task jitter $\Delta I_{avg} \quad = I_{avg} - I'_{avg}$

$$= \frac{\Delta T \times I_{pk}}{T_{dimming}}$$

The change in output current due to dimming task jitter depends on the jitter time and the dimming time period. Increasing the dimming frequency makes the output more susceptible to task jitter in dimming. Hence lower dimming frequencies are preferred for reducing the impact for dimming task jitter. The dimming frequency cannot be below 60 Hz as visual flicker must be avoided.

PWM dimming introduces large low frequency swing in input current requirements. Each channel of LED driver switches between peak current and zero at the dimming frequency. By having the dimming signals of the different channels out of phase with each other the current swing can be reduced. The simplest method is to have a phase difference of 120 degrees between any two dimming signals of the three channels. But this approach is best suited when the current requirements and dimming levels of all the channels are the same. If the current requirements and dimming levels of the channels vary over a wide range this approach does not result in the best distribution of current load. This is demonstrated with a three channel LED driver with some examples.

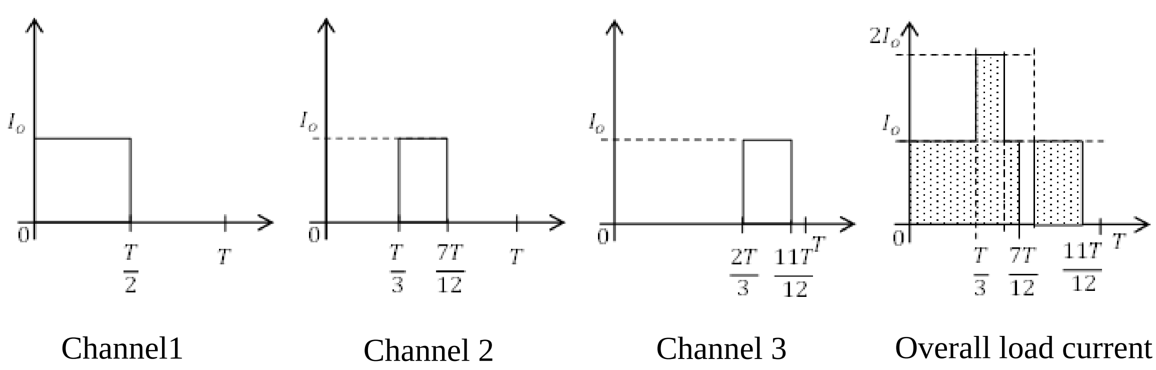

Figure 3-4. Load currents when peak current are identical and the dimming signals are out of phase by 120 degrees

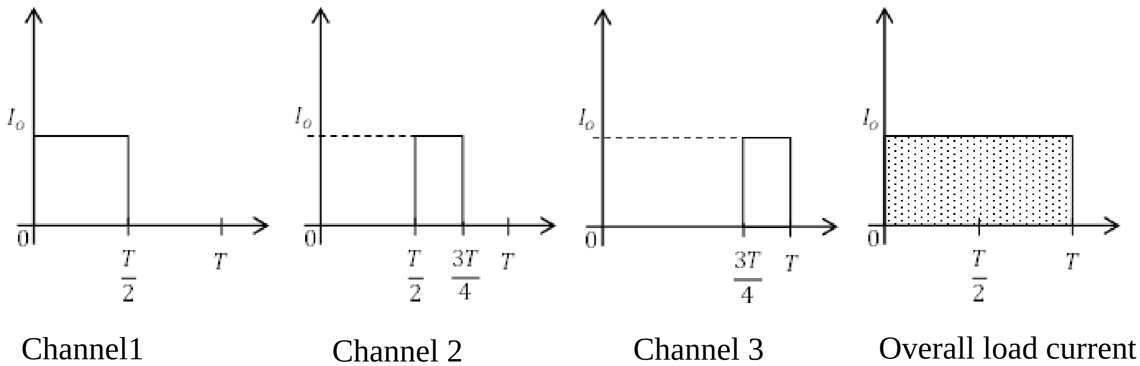

Figure 3-5. Load currents when peak currents are same and dimming signals are optimally set

In Figure 3-4 and Figure 3-5 all three channels have the same peak load currents but different dimming levels. Figure 3-4 shows the current distribution over the dimming period when the dimming signals are spaced 120 degrees out of phase with each other. The distribution in Figure 3-5 is better than the one in Figure 3-4.

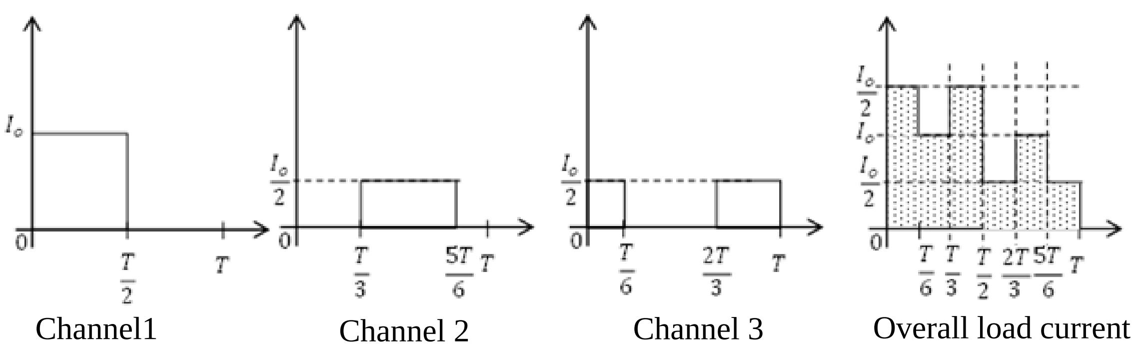

Figure 3-6. Load currents when dimming levels are identical and dimming signals are out of phase by 120 degrees

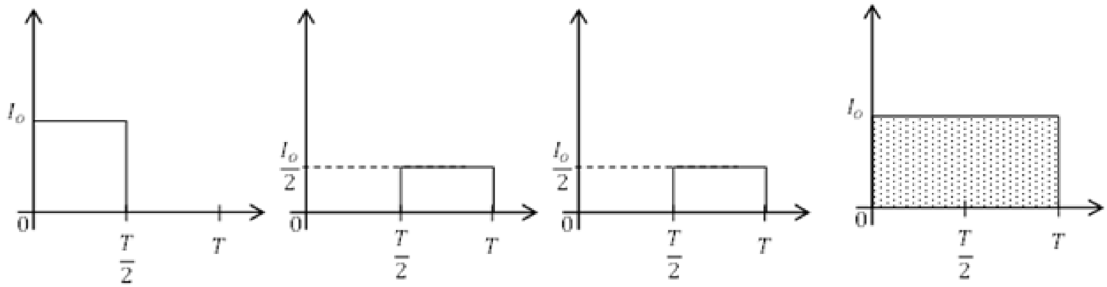

Figure 3-7. Load currents when dimming levels are identical and dimming signals are optimally set.

In Figure 3-6 and Figure 3-7 all three channels have the same dimming levels but different peak load currents. Figure 3-6 shows the current distribution over the dimming period when the dimming signals are spaced 120 degrees out of phase with each other. The distribution in Figure 3-7 is better than the one in Figure 3-6.

In order to reduce the maximum step in load current requirement and reduce the ripple $(I_{max} - I_{min})$we use a more efficient method. Following are the steps of this method:

1. List the channels in ascending order of peak load currents. Channel 1 has the highest peak current requirement and channel 3 has the lowest peak current requirement.
2. Channel 1 is switched on for the duration specified by the dimming level beginning from the start of the dimming time period.
3. Channel 2 is switched on for the duration specified by the dimming level beginning from point when Channel 1 is switched off. If the dimming time period is exceeded start over from the beginning the dimming time period.
4. Channel 3 is switched between the end of dimming time period and point specified by the dimming level of channel 3.

$T_{dimming}$is the dimming time period. $T_{on_1}$and $T_{on_2}$are the durations for which the channels 1, 2 and 3 are switched on respectively. The current load is distributed between 0 and $T_{dimming}$as follows:

1. Channel 1 is on from 0 to T_{on_1}.
2. Channel 2 is on from$T_{on_1}$to $((T_{dimming} + T_{on_2}) mod\ T_{dimming})$.
3. Channel 3 is on from $(T_{dimming} - T_{on_3})$to$T_{dimming}$.

Waveforms for load current requirement given by this method for example 1 and example 2 are represented in Figure 3-5 and Figure 3-7. These distributions have lowest possible maximum current step and lowest ripple $(I_{max} - I_{min})$.

CHAPTER 4

4 Implementation of HB LED driver

4.1 Hardware Implementation

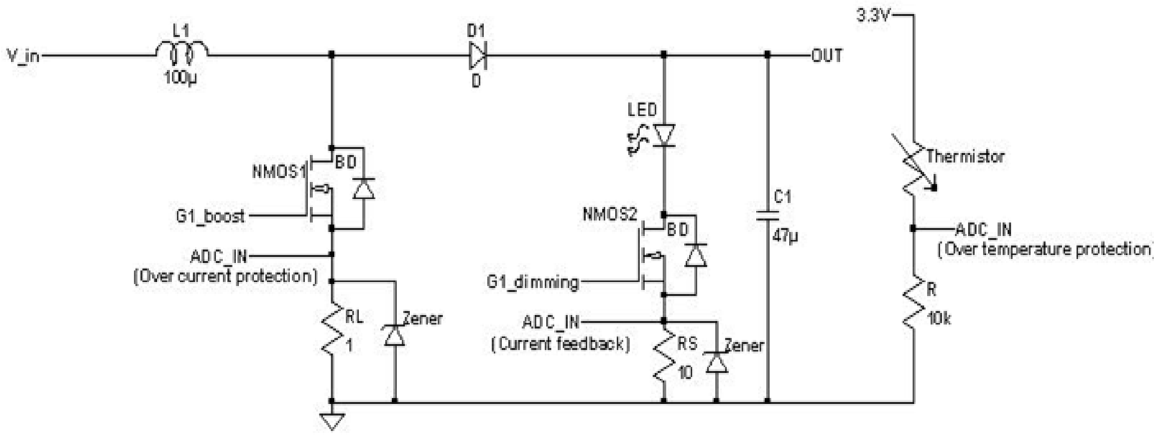

Figure 4-1. Schematic for a single channel of the three channel HB LED driver

4.1.1 Current sensing

Current sensing is required for feedback and over current protection. A 1Ω resistor is used as the current sense resistor. This resistor can be used to detect currents up to 3.3A.

$$I = V_{adc} \times 1$$

Zener diodes are used to protect the microcontroller ADC channels. They prevent the input voltage to the ADC channels from exceeding 3.3 V.

4.1.2 Over temperature protection

A thermistor is attached to the heat sink to monitor the temperature and detect over temperature conditions. The thermistor has the following characteristic. The thermistor has a resistance of 10.30 k Ω at 71F and has a negative coefficient of 0.67 F/k Ω. It is connected in series with a 10 k Ω resistor as shown in Figure 4-1.

The temperature is measured by recording the voltage drop across the 10 k Ω resistor. An ADC channel of the microcontroller is used for this purpose.

4.1.3 Dimming

PWM dimming requires a dimming gate which toggles the load current on and off. An n-channel MOSFET is used as a dimming switch for each of the HB LED loads. The gate terminal of this NMOS is driven by the microcontroller.

4.2 Software Implementation

4.2.1 Initializations

The software first initializes the switching frequency, control loop, dimming control over current protection and over temperature protection parameters. Following are the functions used to initialize the parameters for the HB LED driver.

1 Init_switching_fequency(switching_time_period):

This function is used to set the time period of the boost gate switching signal in terms of numbers of cycles of 31.25ns.

2 Init_control_loop(control_loop_period, Kp_1, Kp_2, Kp_3, Ki_1, Ki_2, Ki_3, Kd_1, Kd_2, Kd_3):

This function sets the control loop frequency by passing the time period of the control loop in terms of number of 31.25ns intervals. The Proportional-Integral-Derivative controller parameters are set for each of the three boost channels.

3 Init_dimming(dim_level_1, dim_level_2, dim_level_3):

This function sets the initial dimming levels for the three channels of the HB LED driver.

4 Init_overcurrent_protection(over_current_1, over_current_2, over_current_3):

This function sets the currents at which the over-current protection is triggered for each of the three channels.

5 Init_temperature_protection(temperature_level_1, temperature_level_2):

This function sets the temperature levels in Fahrenheit at which the over temperature protection is triggered. ‘temperature_level_1’ triggers the lowering of dimming level and ‘temperature_level_2’ triggers the shutdown of the boost converter channel.

4.2.2 Timers

The tasks five tasks recognized in chapter 3 are implemented using the timer interrupts of rl78. The 128-pin rl78 has 8 timer channels. PWM signaling is implemented in rl78 by master-slave combination of timers. The master timer determines the signal time period and the slave timer determines the ‘on time’ of the PWM signal. Multiple slaves can be driven by a single master channel as long as certain conditions are met. Channel 0 is used as the master timer for the boost converter’s gate signal. This is set by the switching frequency of the boost gate. Timer channels channel 2; channel 4 and channel 6 are set as the slave channels of channel 0. The outputs of these slave channels drive the boost gates of HB LED driver channels CH1, CH2 and CH3 respectively. The timer registers of the slave channels are set by the ‘on time’ determined by the control loop of each of the three HB LED driver channels.

The timer channel outputs of channel 1, channel 2 and channel 3 are used for dimming signals of the three HB LED loads. These pins are set as GPIOs and are set high or low based on the dimming levels of their respective channels.

Timer channel 8 is used for running the interrupt for control loop task execution. This timer interrupt is set to trigger at the control loop frequency. The interrupt service routine of this timer has the control loop computations.

Since the number of timers is limited some of the tasks have to be implemented by software triggers. The dimming, overcurrent protection and over temperature protection tasks are triggered by software counts incremented by the control loop interrupt. Hence the time periods of dimming task, over current protection task and over temperature protection task are a multiple of the control task time period.

Priorities for the timer interrupts driving the boost gate signals are set high and control loop interrupt priority is set low.

4.2.3 Analog to digital converters

The rl78 microcontroller has 12 channels for ADC conversion. Each channel of the HB LED driver needs three ADC conversion inputs. One each for measuring the following analog signals:

1. Load current for feedback.
2. Inductor current for overcurrent protection.
3. Voltage drop in the thermistor bridge for over temperature protection.

For three channel HB LED driver a total of 9 ADC channels are required. Table 4-1 shows the ADC channels employed for different analog signals.

Table 4-1. Analog channels of rl78 microcontroller

Analog input	Analog signal measured
ANI0	Channel 1 current feedback
ANI1	Channel 2 current feedback
ANI2	Channel 3 current feedback
ANI3	Channel 1 over temperature protection
ANI4	Channel 2 over temperature protection
ANI5	Channel 3 over temperature protection
ANI6	Channel 1 overcurrent protection
ANI7	Channel 2 overcurrent protection
ANI16	Channel 3 overcurrent protection

4.2.4 Software optimizations

Several optimizations are implemented to reduce the computational cycles, stack size and the code size required to run the software. Lowering the computational cycles allows the control loop to run at a higher frequency and this in turn improves the response time of the closed loop. Reducing the stack size, code size and computational workload also allows estimating the minimum computational power demanded of a microcontroller to effectively run the software. Using a microcontroller with low operating frequency, low stack size and low programmable memory reduces the cost of the LED driver. Following are the optimizations implemented to achieve these goals.

1. The control equations operate on the number of ‘on time’ cycles of the boost gate signals rather than the duty cycle. This eliminates the use of several multiplication and division operations that were used to convert duty cycle to ‘on time’ cycles for timer register. As divisions and multiplication operations require multiple cycles in simple microcontroller this optimization saves computation cycles. This optimization also has another minor positive effect. The control loop computations are free of approximations that occur when duty cycle is converted into ‘on time’ cycles.
2. There are several analog to digital conversions in the software. The time for each conversion is 7.8us when the ADC is operated a maximum frequency. During this conversion time instructions independent of the ADC conversion result can be executed. This optimization eliminates the ADC conversion overhead and reduces the compute cycles.
3. Multiplication and division are cycle consuming operations. In order to avoid some of these operations the constants being multiplied or divided with are set to the nearest power of 2 values. This allows replacing the multiplication with left shift operation and division with right shift operation. For example the number of dimming levels is set as 16, 32 or 64. This optimization further reduces the compute cycles.

CHAPTER 5

5 Results and Analysis

5.1 Preliminary analysis

In order to demonstrate the importance of using the optimum microcontroller we first present the cost breakup of the three channel HB LED driver board. Figure 5-1 provides the cost distribution of the HB LED driver. The components part number and their prices are elaborated in Appendix A.

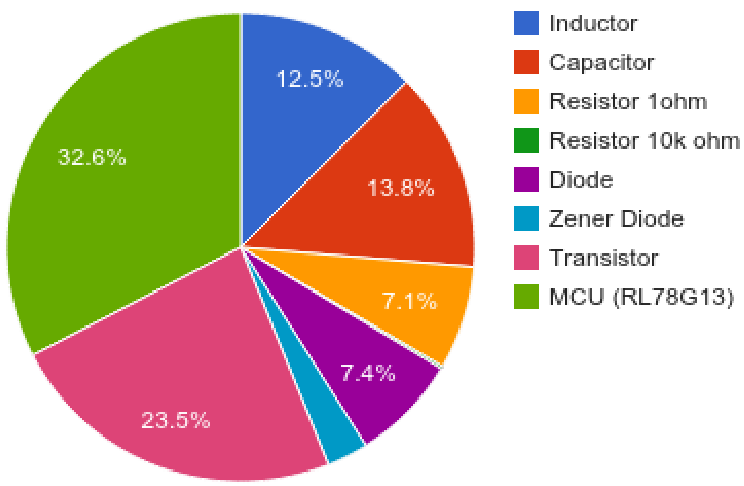

Figure 5-1. Cost distribution of three channel HB LED driver

Figure 5-1shows that the cost of microcontroller has the greatest impact on the overall cost of the HB LED driver. Capacitor and inductor cost have the third and fourth largest share respectively. They have a combined share of 58.9%. Hence in order to bring down the cost of the HB LED driver the MCU, capacitor and inductor combination must be optimized. Section 5.3 presents alternate microcontrollers from the RL78 family, their features and their cost. It is clear from this that using a microcontroller with smaller RAM, smaller memory and slower clock will significantly reduce the cost share of the MCU in the HB LED driver.

As the MCU has the largest share in the HB LED driver cost (32.6%) this will reduce the overall cost of the driver significantly.

5.2 Results

Figure 5-2. Experimental setup: Three channel boost converter daughter board mounted on a RL78G13 development kit

5.2.1 Functionality

The experimental setup has the following parameters:

Switching frequency	= 62.5 kHz	
Control loop frequency	= 1 kHz	
Input voltage	= 2.5 V	
Load current (Channel 1)	= 64 mA	LED string
Load current (Channel 2)	= 39 mA	LCD backlight LED
Load current (Channel 3)	= 258 mA	HB LED

The dimming levels are set such that the load current requirements of the three channels are out of phase with each other. The duty cycles resulting from the closed loop control are 78, 29 and 51 for channels one to three respectively. Figure 5-3, Figure 5-4 and Figure 5-5 are the plots of load currents and gate signals of the three channels.

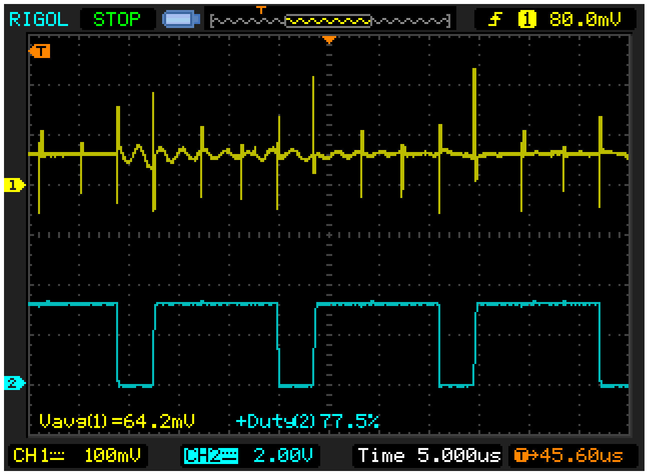

Figure 5-3. CH1: Load current through channel 1 (I_1= 64.2mA) measured across 1 Ω sense resistor; CH2: Boost gate signal of channel 1 (Duty Cycle= 77.5)

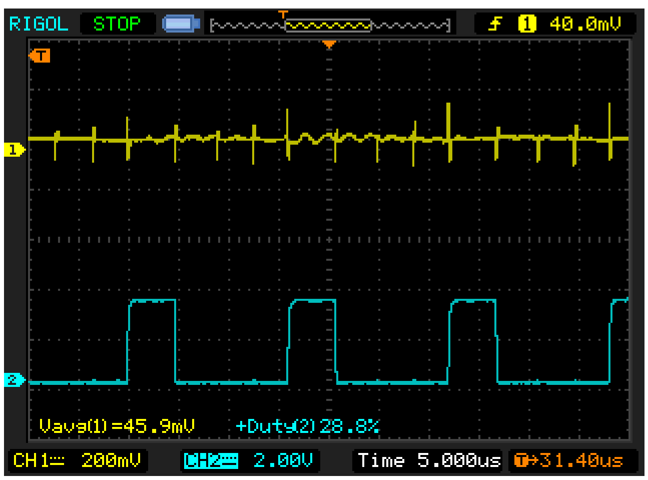

Figure 5-4. CH1: Load current through channel 2 (I_2= 46mA) measured across 1 Ω sense resistor; CH2: Boost gate signal of channel 2 (Duty Cycle= 28.8).

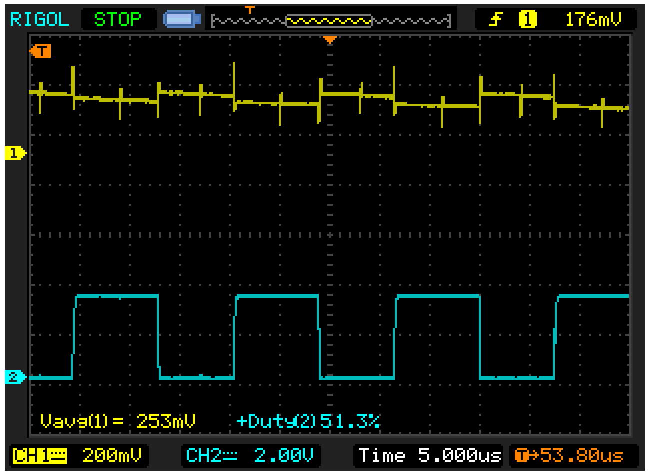

Figure 5-5. CH1: Load current through channel 3 (I_3= 253mA) measured across 1 Ω sense resistor; CH2: Boost gate signal of channel 3 (Duty Cycle= 51.3)

The control loop operation is demonstrated in Figure 5-6. For a step change in load current requirement in channel 3 the control loop operation adjusts the duty cycle of channel 3. The response time is approximately 10ms.

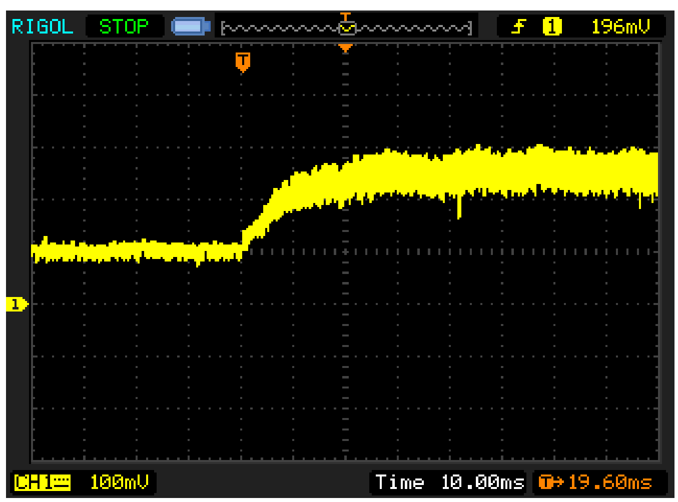

Figure 5-6. Closed loop step response

During the dimming off time the boost gate and the control loop signal are switched off. Figure 5-7 and Figure 5-8 show the dimming operation.

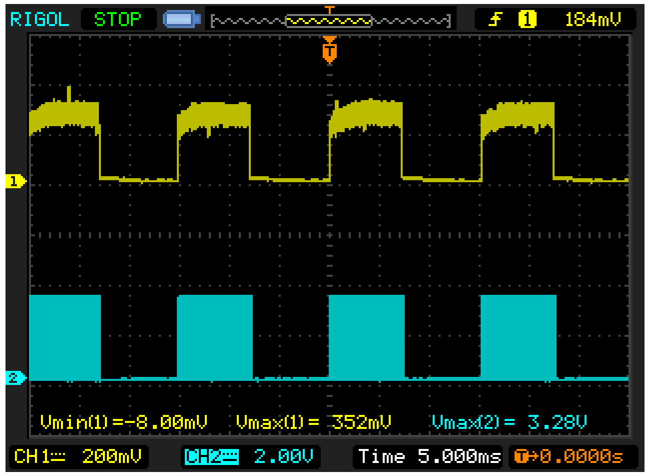

Figure 5-7. CH1: Load current through channel3; CH2: Boost gate signal of channel 3

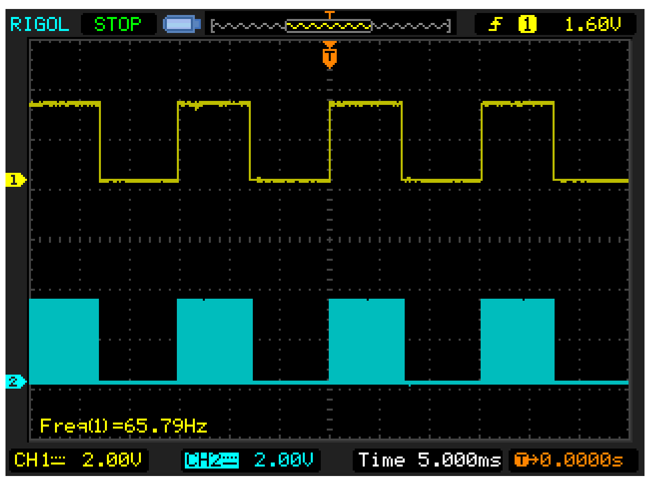

Figure 5-8. CH1: Dimming signal of channel 3; CH2: Boost gate signal of channel 3

The computational load is quantized by the control loop operation signal. This signal is pulled low at the start of the control computations and at the end of the computation it goes back to high. The duration for which this signal is low gives the computation time required for control loop operations. The computational load varies over the dimming period as the control operations are executed for only those channels whose dimming is 'on'. This is demonstrated in Figure 5-9 and Figure 5-10. Maximum time required for control loop computation is 580 μs as demonstrated in Figure 5-11. This occurs when all three channels are driving current.

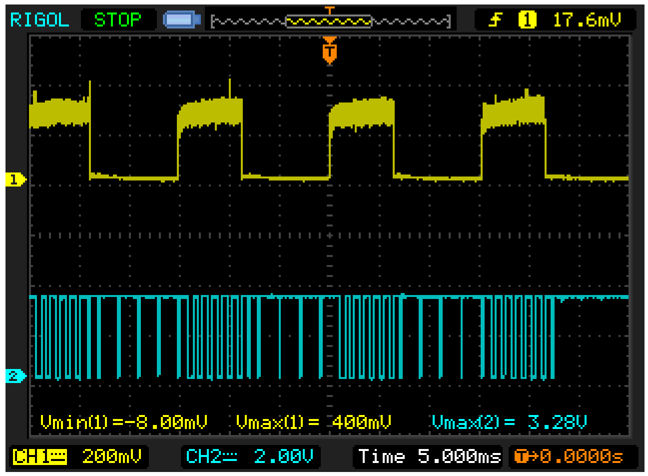

Figure 5-9. CH1: Load current through channel 3; CH2: Control loop operation signal

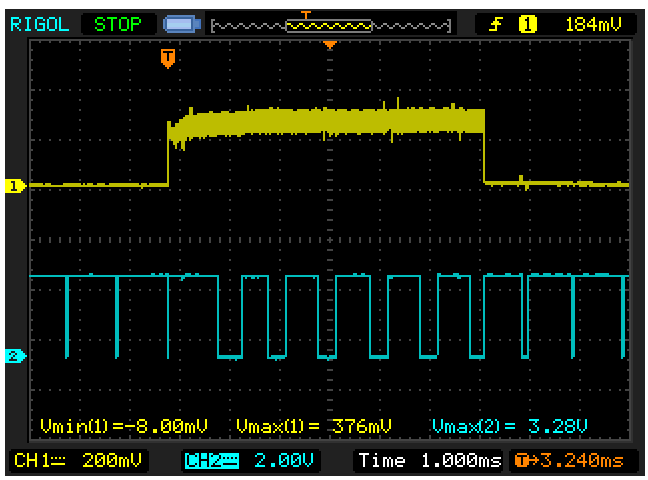

Figure 5-10. CH1: Load current through channel 3; CH2: Control loop operation signal

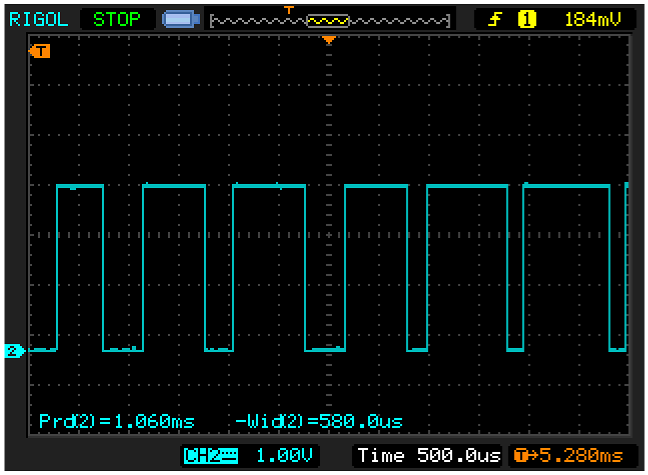

Figure 5-11. CH1: Control loop operation signal. Maximum low period is 580 μs

5.3 Analysis

The software running on the MCU can be optimized with emphasis on saving computational time, memory or RAM. Reducing the memory and RAM requirements for the MCU will reduce the cost substantially. Lowering the computational time requirement allows use of MCU with slower clock and hence lowers cost. The RL78 family of microcontrollers has a wide range of cost. The features offered by this MCU family are presented in Table 5-1. The part number list of microcontrollers selected is in Appendix B. Figure 5-12 shows the variation in the overall cost of HB LED driver when different microcontrollers of this family are used.

Table 5-1. RL78 family microcontrollers: Features and cost

#	Clock (MHz)	RAM (KB)	Memory (KB)	ADC channels	# I/O	Unit cost for 1,000 ($)	Board cost ($)
1	24	0.5	4	11	22	0.7395	4989.54
2	24	1	12	11	22	1.015	5265.04
3	24	1.5	16	11	22	1.0875	5337.54
4	32	2	16	9	28	1.498	5748.04
5	32	2	32	12	38	1.792	6042.04
6	32	4	64	12	38	2.058	6308.04
7	32	8	96	12	38	2.254	6504.04
8	32	23	512	12	38	3.915	8165.04

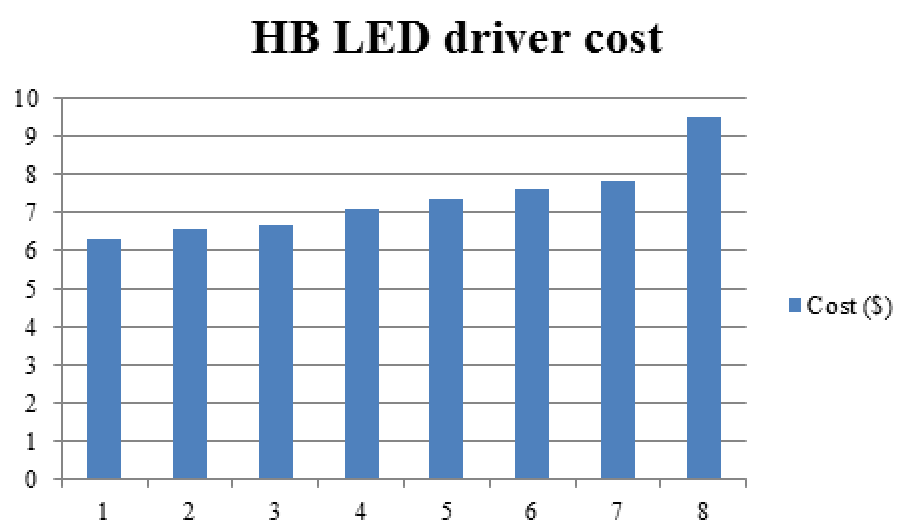

Figure 5-12. Variation of HB LED driver cost with the choice of microcontroller

5.3.1 Impact of optimizing computational time

The MCU of the HB LED driver spends most of its time in control loop operations. The minimum clock frequency ($f_{MCU_{min}}$) required depends on the control loop frequency (f_c) and the compute cycles required for control loop executions (K_c).

$$f_{MCU_{min}} \geq K_c \times f_c$$

This condition ensures that the MCU is able to complete one set of control loop operations before the next control loop iteration. The relationship between control loop frequency (f_c), gate switching frequency (f_{sw}) and capacitance (C) required for stability of PID feedback control is explored in [6].

$$f_c < \alpha \times f_{sw}$$

$$C \gg \frac{1}{2\pi R_{ref} \alpha f_{sw}}$$

α can be maximum of 1/6

R_{ref} is desired closed loop impedance

For our analysis we take $\alpha = 1/50$ and $R_{ref} = 30\Omega$. We use following the equations for the computation of control loop frequency and capacitance.

$$f_c = (1/50) \times f_{sw}$$

$$C = 10 \times \frac{1}{2\pi \times 30 \times (1/50) \times f_{sw}}$$

The inductance value is related to the switching frequency by following equation.

$$L = \frac{V_{in} \times (V_{out} - V_{in})}{f_{sw} \times V_{out} \times \Delta I}$$

Hence the inductance, capacitance and microcontroller frequency are interrelated through the switching frequency. Using a larger capacitor and inductor will require a slower MCU and using a smaller capacitor and inductor will require a faster MCU. There is cost tradeoff between capacitor, inductor and MCU that is affected by the compute cycles required (K_c).

K_c = 36480 cycles before any software optimization. We first analyze this cost trade of for this case. Table 4-1contains the cost for inductors, capacitors and MCU for different switching frequencies. The details of part numbers chosen is in appendix B. Figure 5-13 shows the cost tradeoff between capacitor, inductor and MCU. Switching frequency of 30 kHz gives the minimum total cost. This requires a 24MHz microcontroller.

Table 5-2. Inductor, capacitor, MCU cost and total cost for different switching frequencies. (Before optimization)

#	fsw (KHz)	L (uH)	C (uC)	Minimum MCU frequency (MHz)	MCU frequency (MHz)	Inductor cost (cents)	Capacitor cost (cents)	MCU cost (cents)	Total cost (cents)
1	10	625	266	7.296	8	206.40	121.38	67.50	654.11
2	20	313	133	14.592	16	139.97	53.28	67.50	519.58
3	30	209	89	21.888	24	101.28	53.28	67.50	480.89
4	40	157	67	29.184	32	101.28	48.96	130.50	539.57
5	50	125	54	36.48	40	101.28	44.64	401.45	806.20
6	60	105	45	43.776	48	95.04	44.64	463.05	861.56
7	70	90	38	51.072	56	95.04	42.24	776.55	1172.66

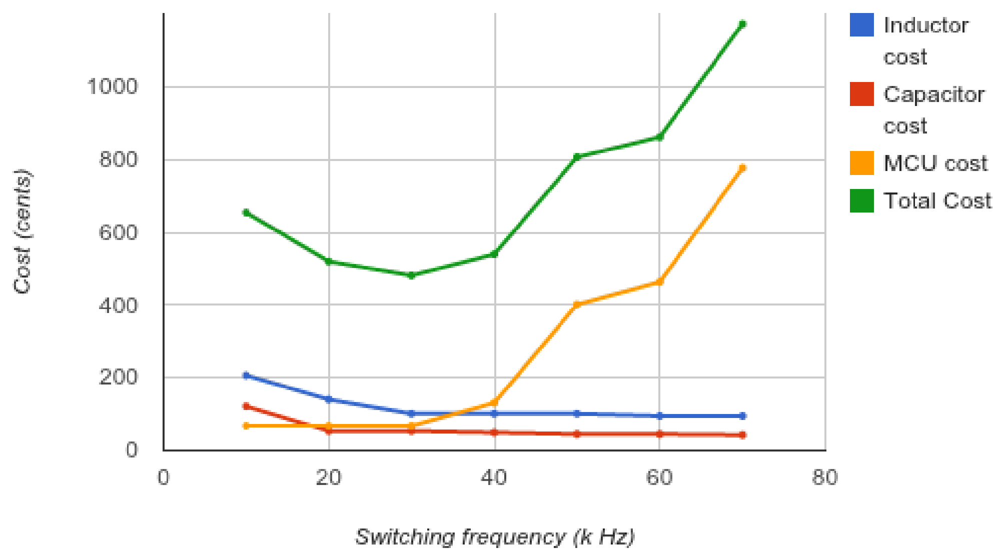

Figure 5-13. Inductor, capacitor, MCU cost and total cost for different switching frequencies. (Before optimization)

After the software optimizations K_c = 18560 cycles. A capacitor-inductor-MCU cost tradeoff analysis for the optimized driver is as follows. Table 5-3 contains the cost for inductors, capacitors and MCU for different switching frequencies. The details of part numbers chosen is in appendix B. Figure 5-14 shows the cost tradeoff between capacitor, inductor and MCU. Switching frequency of 60 kHz gives the minimum total cost. This requires a 24MHz microcontroller too.

Table 5-3. Inductor, capacitor, MCU cost and total cost for different switching frequencies. (After optimization)

#	fsw (KHz)	L (uH)	C (uC)	Minimum MCU frequency (MHz)	MCU frequency (MHz)	Inductor cost (cents)	Capacitor cost (cents)	MCU cost (cents)	Total cost (cents)
1	10	625	266	3.712	4	206.4	121.38	58.50	645.11
2	20	313	133	7.424	8	139.97	53.28	67.50	519.58
3	30	209	89	11.136	16	101.28	53.28	67.50	480.89
4	40	157	67	14.848	16	101.28	48.96	67.50	476.57
5	50	125	54	18.56	24	101.28	44.64	67.50	472.25
6	60	105	45	22.272	24	95.04	44.64	67.50	466.01
7	70	90	38	25.984	32	95.04	42.24	130.50	526.61

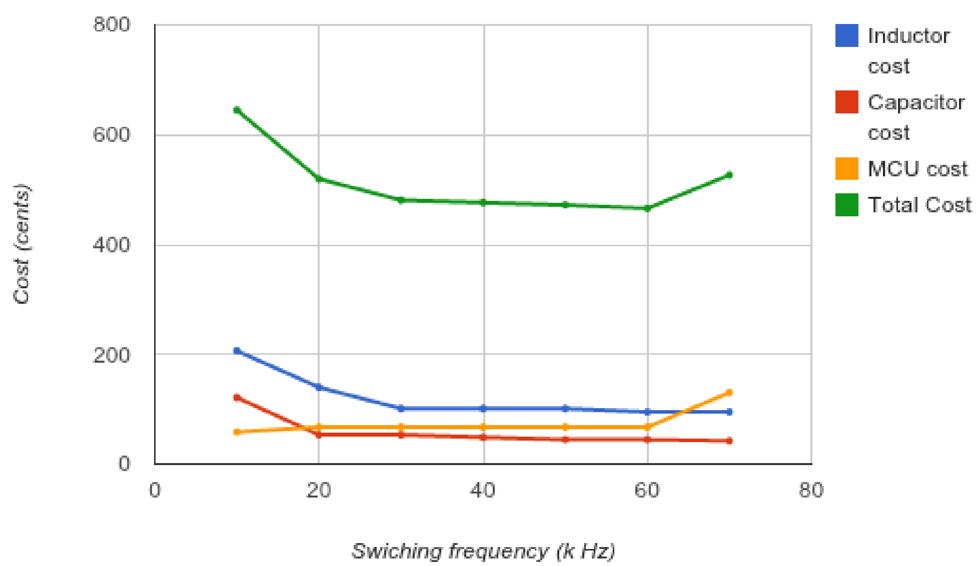

Figure 5-14. Inductor, capacitor, MCU cost and total cost for different switching frequencies. (After optimization)

Optimization of the software for speed reduced the total cost of the driver is reduced by 3.09%.

Another approach to minimize the cost is to have a fixed control loop frequency and reduce the cost of inductor by increasing the switching frequency. In this analysis the stable of the closed loop control is maintained by ensuring that $f_c < (1/6) \times f_{sw}$. Control loop frequency is fixed as 1 kHz. This limits the capacitor size by the following equation:

$$C \gg \frac{1}{2\pi R_{ref} f_c}$$

For this analysis we use capacitor value of $C = 10 \times \frac{1}{2\pi R_{ref} f_c}$ that is 54μF. The minimum required MCU frequency is given by $f_{MCU_{min}} \geq K_c \times f_c$. Before optimization this is 36480 cycles × 1 kHz = 36.48 MHz. The cheapest RL78 MCU with satisfying this criterion has 40 MHz clock (Part Number: UPD78F1201MC-CAB-AX-ND). Table 5-4 presents the total cost variation with increasing switching frequency before optimization.

After the optimizations the minimum required MCU frequency is 18560 cycles × 1 kHz = 18.56 MHz. For this design we can use the cheaper 24MHz RL78 MCU. Table 5-5 presents the total cost variation with increasing switching frequency after optimization. Figure 5-15 shows the variation in cost with increasing switching frequencies for the un-optimized and the optimized designs.

Table 5-4. Inductor, capacitor, MCU cost and total cost for different switching frequencies with fixed control loop frequency (Before optimization)

#	fsw (K Hz)	L (uH)	Inductor cost (cents)	Capacitor cost (cents)	MCU cost (cents)	Total cost (cents)
1	10	625	206.40	44.64	401.45	911.32
2	20	313	139.97	44.64	401.45	844.89
3	30	209	101.28	44.64	401.45	806.20
4	40	157	101.28	44.64	401.45	806.20
5	50	125	101.28	44.64	401.45	806.20
6	60	105	95.04	44.64	401.45	799.96
7	70	90	95.04	44.64	401.45	799.96

Table 5-5. Inductor, capacitor, MCU cost and total cost for different switching frequencies with fixed control loop frequency (Before optimization)

#	fsw (K Hz)	L (uH)	Inductor cost (cents)	Capacitor cost (cents)	MCU cost (cents)	Total cost (cents)
1	10	625	206.40	44.64	67.50	577.37
2	20	313	139.97	44.64	67.50	510.94
3	30	209	101.28	44.64	67.50	472.25
4	40	157	101.28	44.64	67.50	472.25
5	50	125	101.28	44.64	67.50	472.25
6	60	105	95.04	44.64	67.50	466.01
7	70	90	95.04	44.64	67.50	466.01

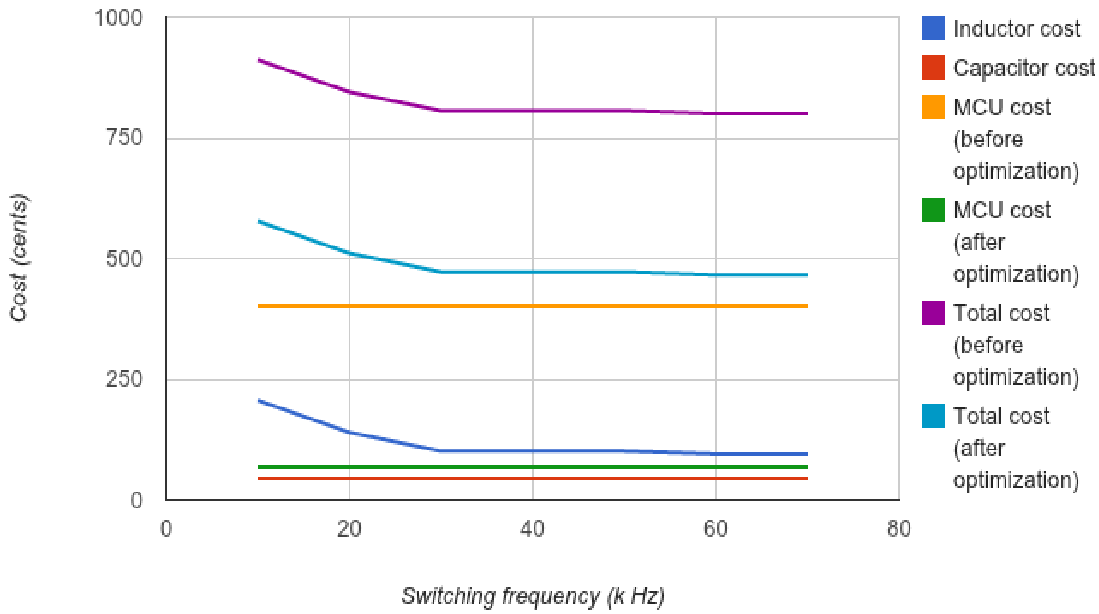

Figure 5-15. Total cost variation for different switching frequencies with fixed control loop frequency (Before and after optimization)

The total cost does not reduce significantly when the switching frequency is increased beyond 30 kHz. Increasing the switching frequency from 10 kHz to 30 kHz reduces the total cost by 11.53% with un-optimized code and 18.21% with optimized code. On the other hand; optimizing the LED driver code for speed allows the use of a significantly cheaper MCU. This reduces the total driver cost decreases by 36.64% to 41.42%.

5.3.2 Impact of optimizing memory and RAM

Reducing the requirements for memory and RAM will not directly affect the inductance and capacitance values. But the cost of the microcontroller is affected by the change in memory and RAM requirements. This impact on the cost is presented in Table 5-6.

Table 5-6. Impact of memory optimization

	RAM (kB) Requirement	Memory (kB) Requirement	RL78 MCU Part number	MCU cost (cents)
Before optimization	3.121	10.202	R5F100EEANA#U0-ND	1.96000
After optimization	0.305	5.693	R5F10368ASP#V0-ND	0.73950

Optimization of the software for memory and RAM reduced the MCU cost by 62.27%. The total cost of the driver goes down by 16.74%.

CHAPTER 6

6 Conclusions and Future Work

6.1 Conclusions

Multiple channel LED driver requires a microcontroller with fast clock. It does not need large memory and RAM sizes. The memory size and RAM size have a large impact on the cost of the microcontroller, hence in order to reduce the cost of the multiple channel HB LED driver the driver code must be optimized for size and a microcontroller with low memory and RAM should be selected. This reduces the cost of the HB LED driver substantially.

With fixed control loop frequency optimizing the driver for speed can lead to significant reduction in cost (24.9% to 29.8%). In this case the increasing the switching frequency will improve the cost up to certain point (8% to 16%) and beyond this the change is not substantial.

If the driver cost is already minimized for a given specifications, the impact of speed optimization is not very significant on the overall cost of the HB LED driver. (2.9%)

Memory optimization can reduce the cost of the HB LED driver substantially (16.74%). Memory optimizations do not affect the cost of capacitor or inductor but will reduce the cost of the microcontroller by a large fraction (62.27%). Hence this optimization will be helpful in designs where microcontroller cost is high.

In some applications the microcontroller is used for multiple tasks. The HB LED driver tasks will have to share resources with other tasks. As the memory requirements for LED driver are low the microcontroller can efficiently execute tasks which require large memory and RAM.

This work shows that there are specific optimizations suited for different designs and that software optimization is an important technique to reduce the total cost of the HB LED driver.

6.2 Future Work

The thermistor used in over temperature control measures the temperature of the heat sink of the HB LED. To derive the junction temperature form this measurement requires thermal modeling of the HB LED. In future work a thermal model is designed for the HB LED.

Future work will also include further optimization of the control loop execution with focus on reducing update rate.

The switching transistors have the second greatest share in the cost of three channel HB LED driver. Finding an optimal transistor will greatly reduce the cost of the driver. Finding alternate transistors is part of future work.

In this research the microcontroller implements the tasks required for HB LED sequentially without any preemption. In many applications HB LED driver is just one of the several modules being monitored by the microcontroller. In these applications microcontroller uses preemptive scheduling. This will impact the cost of the microcontroller and the cost relation among microcontroller, capacitor and inductor. Hence this aspect of cost reduction needs to be explored.

REFERENCES

[1] O. Camacho, *Automotive High Brightness LED- Based on the MC9S0MP16 microcontroller,* Freescale Semiconductor - Application Note, 2010.

[2] D. Chang, B. Larimore and P. Carner, *Methods of controlling multiple LED strings with C2000 MCUs*, Texas Instruments, 2010.

[3] Maxim, *Programmable, Four-String HB LED Driver with Output-Voltage Optimization and Fault Detection,* 2011.

[4] S. Bowling, *A Digital Constant Current Power Driver - AN1138,* Microchip Technology Inc., 2007.

[5] N. N. a. J. P. F. Y. Gu, "White LED performance," in *Proc. 4th Int. Conf. Solid State Lighting,* 2004 .

[6] "Peterchev, A.V.; Sanders, S.R.; , "Load-Line Regulation With Estimated Load-Current Feedforward: Application to Microprocessor Voltage Regulators," Power Electronics, IEEE Transactions on , vol.21, no.6, pp.1704-1717, Nov. 2006".

[7] "www.digikey.com," [Online].

APPENDICES

Appendix A

A.1 Bill of materials

Table 6-1. Bill of materials for three channel HB LED driver (Quantity = 1000)

Component	Vendor	Part Number	Unit price ($)	Quantity	Cost ($)
Inductor (100uH)	Digikey	595-1352-2-ND	0.2635	3000	790.50
Capacitor (47uF)	Digikey	565-2568-2-ND	0.2904	3000	871.20
Resistor (1 ohm)	Digikey	541-10.0AFDKR-ND	0.0750	6000	450.00
Resistor (10k ohm)	Digikey	RMCF0805FT10K0CT-ND	0.0038	3000	11.34
Diode	Digikey	568-6530-2-ND	0.1550	3000	465.00
Zener Diode (3.3 V)	Digikey	MMSZ4684-TPMSCT-ND	0.0300	6000	180.00
Transistor	Digikey	UPA2719AGR-E2-AT-ND	0.2470	6000	1482.00
MCU (RL78G13)	Digikey	R5F101JEAFA#V0-ND	2.0580	1000	2058.00

Appendix B

B.1 List of some RL78 family microcontrollers [7]

Table 6-2. RL78 microcontroller part numbers

#	Microcontroller
1	MCU RL78G12 4KB FLASH 24-WQFN
2	MCU RL78G13 12KB FLASH 24-WQFN
3	MCU 16BIT 16KB FLASH 24WQFN
4	MCU RL78G13 16KB FLASH 40-WQFN
5	MCU 16BIT 32KB FLASH 52LQFP
6	MCU 16BIT 64KB FLASH 52LQFP
7	MCU 16BIT 96KB FLASH 52LQFP
8	MCU 16BIT 512KB FLASH 52LQFP

Appendix C

C.1 List of components (before optimization) [7]:

Table 6-3. Inductor, capacitor, MCU part numbers and cost for different switching frequencies. (Before optimization)

#	fsw (KHz)	L (uH)	C (uC)	Minimum MCU freq. (MHz)	MCU freq (MHz)	Inductor part number	Inductor cost (cents)	Capacitor part number	Capacitor cost (cents)	MCU part number	MCU cost (cents)	Total cost (cents)
1	10	625	266	7.296	8	445-3733-3-ND	206.40	565-2497-2-ND	121.38	R5F10366ASP#V0-ND	67.50	911.32
2	20	313	133	14.592	16	SCRH127/LD-391-ND	139.97	565-2134-2-ND	53.28	R5F10366ASP#V0-ND	67.50	844.89
3	30	209	89	21.888	24	SC108-271-ND	101.28	565-2134-2-ND	53.28	R5F10366ASP#V0-ND	67.50	806.20
4	40	157	67	29.184	32	SC108-221-ND	101.28	565-2495-2-ND	48.96	R5F1016AASP#V0-ND	130.50	806.20
5	50	125	54	36.48	40	SC108-151-ND	101.28	565-2494-2-ND	44.64	UPD78F1201MC-CAB-AX-ND	401.45	806.20
6	60	105	45	43.776	48	SCRH125-121-ND	95.04	565-2494-2-ND	44.64	R5F52105ADFM#V0-ND	463.05	799.96
7	70	90	38	51.072	56	SCRH125-121-ND	95.04	565-2568-2-ND	42.24	UPD70F3453GF-GAS-AX-ND	776.55	799.96

C.2 List of components (after optimization) [7]:

Table 6-4. Inductor, capacitor, MCU part numbers and cost for different switching frequencies. (After optimization)

#	fsw (KHz)	L (uH)	C (uC)	Minimum MCU freq. (MHz)	MCU freq (MHz)	Inductor part number	Inductor cost (cents)	Capacitor part number	Capacitor cost (cents)	MCU part number	MCU cost (cents)	Total cost (cents)
1	10	625	266	3.712	4	445-3733-3-ND	206.40	565-2497-2-ND	121.38	M34286G2GP#U0-ND	58.50	577.37
2	20	313	133	7.424	8	SCRH127/LD-391-ND	139.97	565-2134-2-ND	53.28	R5F10366ASP#V0-ND	67.50	510.94
3	30	209	89	11.136	16	SC108-271-ND	101.28	565-2134-2-ND	53.28	R5F10366ASP#V0-ND	67.50	472.25
4	40	157	67	14.848	16	SC108-221-ND	101.28	565-2495-2-ND	48.96	R5F10366ASP#V0-ND	67.50	472.25
5	50	125	54	18.56	24	SC108-151-ND	101.28	565-2494-2-ND	44.64	R5F10366ASP#V0-ND	67.50	472.25
6	60	105	45	22.272	24	SCRH125-121-ND	95.04	565-2494-2-ND	44.64	R5F10366ASP#V0-ND	67.50	466.01
7	70	90	38	25.984	32	SCRH125-121-ND	95.04	565-2568-2-ND	42.24	R5F1016AASP#V0-ND	130.50	466.01