STUDY OF DC-DC BOOST CONVERTER’S EFFICIENCY USING CAPACITOR AS ENERGY INPUT

CHOONG PAK WAI

A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor (Hons.) of Electrical And Electronic Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2011

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ____________________________

Name : CHOONG PAK WAI________

ID No. : 07UEB07126__ ______________

Date : 12 MAY 2011______________

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"STUDY OF DC-DC BOOST CONVERTER'S EFFICIENCY USING CAPACITOR AS ENERGY INPUT"** was prepared by **CHOONG PAK WAI** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _________________________

Supervisor : Mr. Francis Lau

Date : _________________________

Specially dedicated to

my beloved mother and father

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Mr. Francis Lau for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, there are also other parties that are directly and indirectly involve in the project. Special thanks to Mr. Ng Mow Song who helps and provide me guidance on PIC programming, UTAR EV team for the support of offering their workplace for us to do testing, and also Uncle Ho for the nice effort for fabricating all the PCB.

Here also, I would like to thank my dear family for providing moral as well as monetary support for me to complete the project smoothly.

Finally, to my project partner Foo Shin Loong for the full support and cooperation of working together and getting this done nicely.

STUDY OF DC-DC BOOST CONVERTER'S EFFICIENCY USING CAPACITOR AS ENERGY INPUT

ABSTRACT

The purpose of this study was to determine the characteristics of the DC-DC boost converter when subjected to both resistive and reactive loads such as DC motor. With the limited energy storage onboard an electric vehicle (EV), efficient power converter is needed to prevent any wastage, apart from driving the vehicle.

Here, we are using supercapacitor as the energy storage in our studies. Supercapacitor technology has been steadily advancing with ever increasing energy storage capacity. Hence, the possibility of switching storage technology from lithium-ion battery to supercapacitor is promising in the near future.

Supercapacitor does not maintain a constant output voltage as its energy level reduces. Therefore, a regulator is required to stabilize its output level, and the boost converter is a good candidate for such a requirement. The boost converter is designed so that it can operate at a large range of input voltage to maximize the energy transfer from the capacitor. Our study is to determine the best range that produces the best efficiency.

TABLE OF CONTENTS

DECLARATION	**ii**
APPROVAL FOR SUBMISSION	**iii**
ACKNOWLEDGEMENTS	**vi**
ABSTRACT	**vii**
TABLE OF CONTENTS	**viii**
LIST OF TABLES	**xi**
LIST OF FIGURES	**xii**
LIST OF SYMBOLS / ABBREVIATIONS	**xvi**
LIST OF APPENDICES	**xviii**

CHAPTER

1	**INTRODUCTION**			**1**
	1.1	Background		1
	1.2	Aims and Objectives		2
2	**LITERATURE REVIEW**			**3**
	2.1	Energy Storing Devices		3
		2.1.1	Introduction	3
		2.1.2	Batteries	3
		2.1.3	Capacitors	4
		2.1.4	Comparison between the Energy Storage	5
	2.2	Power Electronic		8
		2.2.1	Converter Introduction	8
		2.2.2	DC Steps Up Converter	10

		2.2.3 Design and Calculation	11
		2.2.4 Performance	12
	2.3	Measuring Devices	12
		2.3.1 System Efficiency	12
		2.3.2 Digital Voltage Measuring	13
		2.3.3 Digital Current Measuring	16
		2.3.4 Microcontroller Unit	19
	2.4	Electric Vehicle	20
		2.4.1 Background	20
		2.4.2 EV Systems and Subsystems	21
		2.4.3 Energy Source Comparison	22
3	**METHODOLOGY**		**24**
	3.1	Introduction	24
	3.2	Job Division	25
	3.3	Efficiency Measuring Device	26
		3.3.1 Microcontroller	26
	3.4	Power Sensing Device	27
		3.4.1 Voltage Measurement	27
		3.4.2 Current Measurement	29
	3.5	ADC sampling	31
	3.6	Data Display and Transfer	32
	3.7	Program Code Architecture	35
4	**RESULTS AND DISCUSSIONS**		**37**
	4.1	Digital Voltmeter (DVM)	37
		4.1.1 Loading Effect	37
		4.1.2 Fluctuation of Sampled Value	39
	4.2	Accuracy of DVM	40
	4.3	Digital Am Meter (DAM)	43
		4.3.1 Boost Converter Input Side	43
		4.3.2 Boost Converter Output Side	46
	4.4	Real Time Monitoring Using Fixed Load	47

		4.4.1 Boost Converter Efficiency	47
		4.4.2 Boost Converter Auto-Shutdown	49
	4.5	Real Time Monitoring For Motor	50
		4.5.1 Motor Efficiency Monitoring	50
		4.5.2 Loading Effect of Measurement Device	52
	4.6	Capacitor Performance	54
		4.6.1 Supercapacitor Energy Density	56
5	**CONCLUSION AND RECOMMENDATIONS**		**57**
	5.1	Conclusion	57
	5.2	Recommendation	58
	5.3	Future Direction	59
		5.3.1 Current Technology	59
		5.3.2 Future Technology	60
REFERENCES			**62**
APPENDICES			**64**

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Batteries and Capacitor comparison	7
2.2	ADC Module Specification	15
2.3	Advantage and Disadvantage of Current Sensing Mode	17
2.4	Battery versus Ultra-Capacitor Performances	22
2.5	Natural Source Energy Specification	22
3.1	Job Division	25
4.1	Voltmeter analysis for input side	41
4.2	Voltemeter analysis for output voltage	42
4.3	Input Current Data Sampled Analysis	45
4.4	Summarize of Real Time Simulation	48
4.5	Energy Conversion for Capacitor Performance	55
4.6	Supercapacitor Energy Density	56

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	Ragone Plot of Electrochemical Devices	6
2.2	Comparison of voltage conversion	8
2.3	DC to DC Flyback Converter	9
2.4	DC DC Boost Converter Circuit	11
2.5	ADC Sampling	14
2.6	Pi Attenuator	16
2.7	T Attentuator	16
2.8	Low Side Current Sensing	17
2.9	High Side Current Sensing	17
2.10	Instrumental Amplifier Circuit	18
2.11	High Side Current Sensing Differential Circuit	19
2.12	General electrical configuration of EVs	21
3.1	Microcontroller Basic Configuration	27
3.2	Attenuation Using Voltage Divider	28
3.3	Rsense using Inverting Amplifier	30
3.4	Rsense using Non-Inverting Amplifier	30
3.5	4 Bits LCD Module Interconnection	32
3.6	UART Module Interconnection	33
3.7	UART tools provided from PICkit 2 software	34

3.8	Log File copied from UART Tools	34
3.9	Data is converted into excel format using the import wizard	35
4.1	Internal ADC Structure	38
4.2	Loading Effect Solution	38
4.3	Jitter in clock signal that degrades the ADC signal-to-noise ratio.	40
4.4	Amplitude change causes by jitter	40
4.5	Input current at low PWM duty ratio	44
4.6	Input current at high PWM duty ratio	44
4.7	Output Current low PWM duty ratio	46
4.8	Output Current high PWM duty ratio	46
4.9	Discharging Duration with Timer Malfunction	49
4.10	Timer Frequency Doubler	50
4.11	Motor Input Current Operating Range	51
4.12	Motor Efficiency Operating Range	51
4.13	Discharge duration with Rsense Loading	53
4.14	Discharge duration without Rsense Loading	53
4.15	Tested Boost Converter Efficiency with Varying Current	54
5.1.1	Resistorless Current Sensing	58
A.1	Input current for load at 25 Ω	64
A.2	Input Voltage for load at 25 Ω	65
A.3	Output Current for load at 25 Ω	65
A.4	Output Voltage for load at 25 Ω	66
A.5	Efficiency for load at 25 Ω	66

A.6	Input Current for load at 50 Ω	67
A.7	Input Voltage for load at 50 Ω	67
A.8	Output Current for load at 50 Ω	68
A.9	Output Voltage for load at 50 Ω	68
A.10	Efficiency for load at 50 Ω	69
A.11	Input Current for load at 100 Ω	69
A.12	Input Voltage for load at 100 Ω	70
A.13	Output Current for load at 100 Ω	70
A.14	Output Voltage for load at 100 Ω	71
A.15	Efficiency for load at 100 Ω	71
A.16	Input Current for load at 220 Ω	72
A.17	Input Voltage for load at 220 Ω	72
A.18	Output Current for load at 220 Ω	73
A.19	Output Voltage for load at 220 Ω	73
A.20	Efficiency for load at 220 Ω	74
A.21	Input Current for load at 440 Ω	74
A.22	Input Voltage for load at 440 Ω	75
A.23	Output Current for load at 440 Ω	75
A.24	Output Voltage for load at 440 Ω	76
A.25	Efficiency for load at 440 Ω	76
A.26	Input Current for Motor running at freewheel	77
A.27	Input Voltage for Motor running at freewheel	77
A.28	Output Current for Motor running at freewheel	78
A.29	Output Voltage for Motor running at freewheel	78
A.30	Efficiency for Motor running at freewheel	79

B.1	Schematic for microcontroller	80
B.2	Microcontroller’s PCB	81
B.3	Schematic for attenuation using voltage divider	81
B.4	Attenuation and Buffer PCB	82
B.5	Current Sensing Schematic	82
B.6	Current Sensing PCB	83

LIST OF SYMBOLS / ABBREVIATIONS

C_{HOLD}	capacitor voltage reading for ADC, Farad
I_{IN}	input current, Amp
I_{OUT}	output current, Amp
$M_{V\,DC}$	DC voltage transfer function, V_O/V_I
P_{OUT}	output power, W
P_{IN}	input power, W
R_F	diode forward resistance, ohm
R_L	load resistance, ohm
V	applied voltage, Volt
V_F	threshold voltage of diode, Volt
V_C	capacitor voltage, Volt
V_I	input voltage, Volt
V_O	output voltage, Volt
V_{ref}	reference voltage with reference to ground, Volt
$V_{(ref +)}$	+ve reference voltage for ADC, Volt
$V_{(ref -)}$	-ve reference voltage for ADC, Volt
V_∞	steady state voltage, Volt
W	power, Watt

AC	Alternating current
ADC	Analogue to digital conversion
CCM	Continuous conduction mode
DAM	Digital Ammeter
DCM	Discontinuous conduction mode

DVM	Digital Voltmeter
EDLC	Electric Double Layer Capacitor
EV	Electric Vehicle
LIB	Li-ion Battery
LIC	Li-ion Capacitor
PCB	Printable Circuit Board
PWM	Pulse Width Modulation
RTOS	Real Time Operating System
RS232	Serial Communication Standard
UART	Universal Asynchronous Receiver Transmitter

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Graphs	64
B	PCB schematic	80
C	Program code	84

CHAPTER 1

INTRODUCTION

1.1 Background

On average, a conventional combustion engines uses 15% of energy conversion from the fuel. On the other hand, electric vehicle (EV) is able to yield efficiency of around 80% from the energy storage (Shah, 2009). The potential of energy saving through EV is there, as is already been proven by engineers who developed the EV.

The standard of such as high value of efficiency as been proven theoratically. Power electronics component which uses switching regulator instead of linear regulator has dramatically improve the overall efficiency to above 90%. (Kazimierczuk, 2008)

Till present, batteries still dominate the market of energy storing devices. However, it is known that batteries will cause harmful chemical leakage over time. Besides that, the low recharging cycle life of it may not be so suitable for application such as EV, where recharging is expected to be done more frequent.

Hence, a new proposed solution of using supercapacitor, electrolytic double layer capacitor (EDLC) as the energy storing device may be consider as an alternative solution for the low recharging cycle life. With the rapid grow of EDLC technology toward the energy storing density, as well as some new innovation such as lithium doped EDLC (JSR Micro, 2006), it is without doubt that this may be the future substitute for batteries.

1.2 Aims and Objectives

The aim of this project is to study the efficiency of the pulse switching regulator using EDLC as the input source. The goal of it is to provide a stepping stone for future development of EV that uses EDLC as their primary energy storage.

The progress of this project is to succeed the following objective as in providing guideline in the research later.

1. Understand the characteristic of EDLC, such as discharging time, power and energy capacity.
2. Understands the various topologies provided by the switching regulator.
3. Develop a functional switching regulator that can operate at a varying input voltage.
4. Develop a power meter to assist the study of efficiency for the converter.
5. Perform analysis for practical and theoretical conversion result.

CHAPTER 2

LITERATURE REVIEW

2.1 Energy Storing Devices

2.1.1 Introduction

There is various form of energy storing devices available in the market. Battery, capacitor, inductor, flywheel are few of those that has been generally recognisable by engineers. However, in term of storing capability, only batteries have been highly recommended due to its high energy and power density (Battery University, 2005) that allows user to use devices remotely for a longer duration of time.

As far as portable device is concern, energy density is an important criterion to ensure a more durable application on a single charge. On the other hand, power density is the ability to supply burst of energy from the storing device. Here, this kind of burst of energy is useful to give the initial spin for electric motor.

2.1.2 Batteries

In the industry, there are various kind electro-chemical batteries available in the market. Among them, lead-acid, nickel-cadmium, nickel-metal-hydride and lithium-ion are the few of those that are popular and frequently saw and used. This is highly because of one characteristic that beat the likes of alkaline battery which is rechargeable. This is important as industrial maker tries to produce device that is convenience for the consumer to reduce the hassle of purchasing battery.

Lead acid is by far the most established kind of recycle battery available. It has a very low energy-to-weight ratio and a low energy-to-volume ratio. On the other hand, their ability to supply high surge currents and energy storage has made them quite popular with modern application such as car battery, hospital equipment, wheelchairs, emergency lighting and UPS systems.

A lithium-ion battery (LIB) is sometimes known as Li-ion battery. It is generally an electro-chemical battery which the cathode is generally made from one of the three materials. They are a layered oxide such as lithium cobalt oxide, a polyanion such as lithium iron phosphate, and a spinal such as lithium manganese oxide (Hackeray, Thomas, & Whittingham, 2000).

2.1.3 Capacitors

Capacitor is an electronic component that store electrical charge. The charge is stored in the E-Field (Electrical Field) through 2 parallel conducting plates. However, this typical topology is by far good at dealing with application such as bypass capacitor, filter and phase shifter, due to its low capacitance which is range from a few pico farad to few mille farad depending on the type of capacitor used (Radio Electronics, 2010).

Application wise, more capacitance means more energy is stored. And more energy translates to longer durability and life span for device that runs on energy storage. Another advantage of capacitor over battery is the superior recharging cycle without decreasing of performance.

That is until a revolution breakthrough of applying double layer configuration. This gives rise to a new form of capacitor such as electrolytic double-layer capacitor

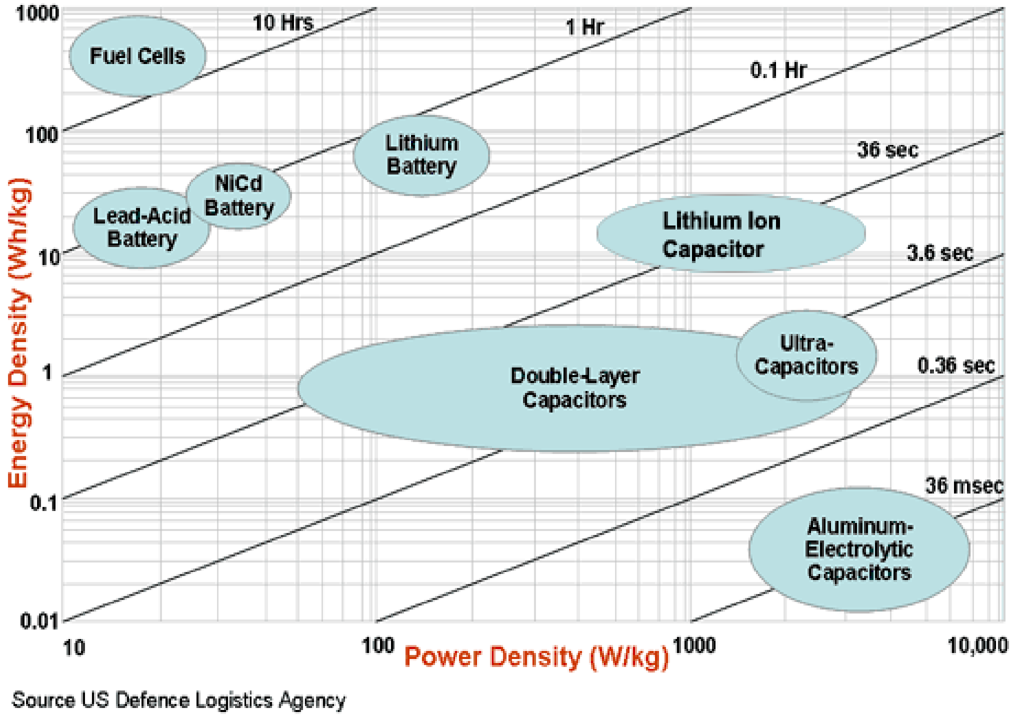
(EDLC), and lithium ion capacitor (LIC) that have a much larger capacitance value. This high capacity capacitor sometimes is referred to as supercapacitor. The capacitance of this new form capacitor has been steadily increased over the years. By 2010, the recorded capacitance for a single capacitor is approximately 3000F (JSR Micro, 2006).

It is also noted that the performance capability of an LIC is far more superior as compared to EDLC in term of energy density. This is highly due to the high working voltage of 3.8V as compared to EDLC of 2.7V.

2.1.4 Comparison between the Energy Storage

There is no defined answer of which energy storage is the best. It is merely depends on the application that is best suit the energy storage that is used. Those factors include size, weight, energy density, power density, reliability, durability, charging cycle, shelf life and etc.

Below is a comparison of the battery and capacitor simplified into Ragone Plot.



(Woodbank Communications Ltd, 2005)

Figure 2.1: Ragone Plot of Electrochemical Devices

As notice from the graph, lithium-ion battery (LIB), lead acid battery, lithium-ion capacitor (LIC), and electric double-layer capacitor (EDLC) are the focus of technology discussion in this section. On average, LIB (8 Wh/kg) has a slight leads over lead acid Battery (1 Wh/kg) and LIC(0.5 Wh/kg) in term of energy density by approximately 7 Wh/kg. In contrast, both the capacitor LIC and EDLC have a far more superior performance in terms of power density. The reading of the power density is 2 kW/kg and 200 W/kg respectively.

Next, is a summary of individual battery and capacitor comparison in the form of table shown below (Battery University, 2005):

Table 2.1: Batteries and Capacitor comparison

	NiCd	**NiMH**	**Lead Acid**	**LIB**	**EDLC**	**LIC**
Energy Density (Wh/kg)	45-80	60-120	30-50	150-190	2-9	12-14
Cycle Life	100-200	200-300	<100	150-300	>100000	>100000
Fast Charge Time	1 Hr	2 -4 Hr	8-16 Hr	1.5-3 Hr	<1 Hr	<1 Hr
Overcharge Tolerance	Moderate	low	high	Low	low	Low
Self-Discharge (per month)	20%	30%	5%	10%	>30%	<2%
Nominal Voltage	1.25	1.25	2.0	3.7	2.7	3.7
Toxicity	Highly Toxic	Low Toxicity	Highly Toxic	Low Toxicity	None	Low Toxicity

NiCd - Nickel-Cadmium
NiMH - Nickel-Metal Hydride
LIB - Lithium-ion Battery
EDLC – Electric Double Layer Capacitor
LIC - Lithium-ion Capacitor

Cycle life is what we known as how many times it is able to charged and discharged while the performance is maintained as good as the initial state. The advantage is clear for both type of capacitor (EDLC and LIC) as it has cycle life over 1000 times more. Besides that, quick charging time is also a plus when application such as EV is considered.

It is well known that capacitor typically have a superior power density as compared to batteries. In contrast, batteries are better in performance when comparing energy density. However, in terms of technology growth, it is well estimated that LIC may soon exceed the performance of LIB. And that this will encourage usage of capacitor to be more applicable to most high power device application.

2.2 Power Electronic

2.2.1 Converter Introduction

Power electronic is the interface where it provides intermediate of efficient conversion between the main power source and the load (Wilson, 2000). Instead of the traditional circuit of using linear conversion, high efficient conversion basically uses switching method to regulate the output. Figure below is a simple diagram that explains the difference in conversion between linear and switching devices.

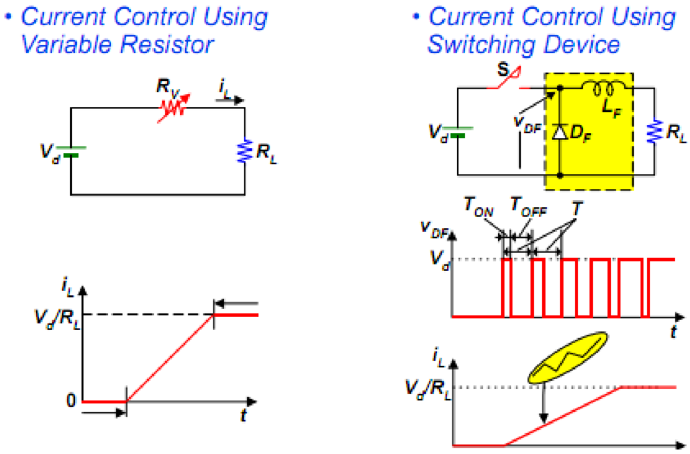
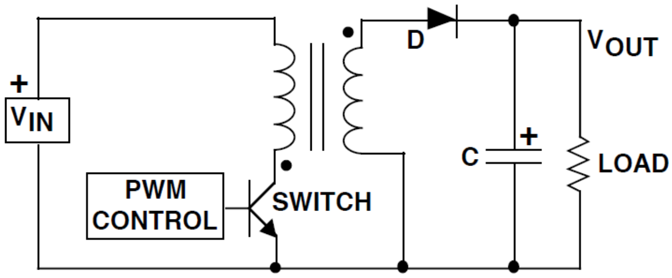


Figure 2.2: Comparison of voltage conversion

Apart from the better efficiency, the advantage of switching converter also include smaller in size and cost. This is largely depending on the size of the transformer used. Either way, switching transformer is generally smaller and cheaper compared to linear transformer.

Besides current source (AC/DC), criteria of designing converter also include whether isolation between input and output. This is normally done by adding switching transformer to isolate them. Flyback Cconverter is one example of DC-DC that uses transformer as isolation.



(Switching Regulator, 2010)

Figure 2.3: DC to DC Flyback Converter

Generally, there are 4 categories that separate switching operation between the input and the output. They are:

- AC-DC Converter (Rectifier)
- AC-AC Converter (Power Controller)
- DC-AC Converter (Inverter)
- DC-DC Converter (DC Chopper - Buck/Boost/Buck-Boost Converter)

Among the widely used basic converter for DC-DC are as below (Switching Regulator, 2010):

- Buck: used to reduce a DC voltage to a lower DC voltage.
- Boost: provides an output voltage that is higher than the input.
- Buck-Boost (invert): an output voltage is generated opposite in polarity to the input.
- Flyback: isolated design, as well as multiple outputs and different polarity.

After understanding the concept of all the available topologies, our focus will on boost topology. According to the application we going to build on as describe in Chapter 3, the output require is around 12V. Mean while, the input is basically being supplied by 2 EDLC at 2.7V rated. Besides that, we assumed that the load is not sensitive; hence isolated topology such as flyback is not required as well.

2.2.2 DC Steps Up Converter

DC-DC switching regulator is rather popular in most of the regulator devices we used nowadays. Its yield higher power conversion efficiency and increased design flexibility which usually consist of 4 major components. They are a Power MOSFET as a controllable switch S, a diodeD_1, an inductorL_1, and a filter capacitor C.

The regulation of the output voltage is controlled by the Duty cycle, D of the pulse-width modulator (PWM) and also the frequency of the PWM. This controls the switching of the power MOSFET which then controls the flow of current in the inductor.

The operation of this switching method to step up or step down basically lies on the current flow by charging and discharging of the inductor. And the time for it to charge or discharge is governed by the power MOSFET, S which acts as a switch via PWM.

In boost converter, the output voltage, V_O is always higher than the input voltage, V_I. Therefore, this is equivalent to a step up transformer for an AC.

However, the boost converter has a poor ability to prevent hazardous transient and failure. For instance, where a voltage surge appears at the input, the diode is turn ON for many cycles. This generate large current spike, which may damage the diode.

Besides using as a dc step up voltage, this converter is also commonly used as an active power factor corrector.

2.2.3 Design and Calculation

To begin, we need to determine the DC voltage transfer function for all the maximum, minimum and nominal values (Kazimierczuk, 2008).

$$M_{V\,DC(max,min,nom)} = \frac{V_O}{V_{I(max,min,nom)}}$$

Then determine all the duty cycle required by determining the converter efficiency

$$D_{(max,min,nom)} = 1 - \frac{\eta}{M_{V\,DC(max,min,nom)}}$$

Next determine the minimum inductance, L_{min} required by assuming a switching frequency, f_S

$$L_{min} = \frac{2}{27}\frac{R_{Lmax}}{f_S}$$

Thus, the current and voltage stresses of the MOSFET and diode are

$$I_{SMmax} = I_{DMmax} = \frac{I_{Omax}}{1 - D_{max}} + \frac{V_O D_{max}(1 - D_{max})}{2 f_s L}$$

And the minimum filter capacitance is

$$V_{Cpp} = V_{rcpp} = \frac{V_r}{2} = \frac{V_O/100}{2}$$

$$C_{min} = \frac{D_{max} V_O}{f_S R_{Lmin} V_{Cpp}}$$

Therefore, the converter efficiency is

$$\eta = \frac{1}{1 + \frac{r_L + D r_{DS}}{(1-D)^2 R_L} + \frac{V_F}{V_O} + \frac{R_F + D r_C}{(1-D) R_L} + f_s C_O R_L}$$

The parameters are as in the diagram below

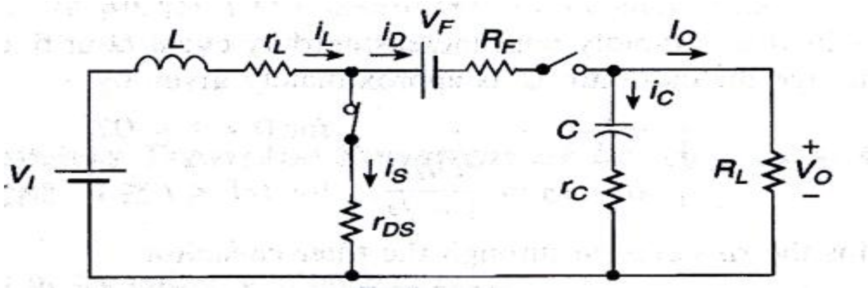


Figure 2.4: DC DC Boost Converter Circuit

2.2.4 Performance

The boost converter is a step up circuit without using transformer. A proper design of the circuit by controlling the components value can allow the efficiency to be higher than 90%.

Practically wise, the value of 90% may be limited up number of factor as stated below:

- Forward voltage and internal resistance of the diode
- Internal resistance of the MOSFET
- Tolerance value of inductance as well as the Q value.
- Switching frequency of MOSFET to on and off

2.3 Measuring Devices

2.3.1 System Efficiency

System efficiency basically tells the effectiveness of a device when subjected to an input. Normally, it is measured in terms of percentage that is energy out over energy in. The term efficiency is applied into all field of calculation from engineering to business. As for electronic devices, such as power electronic devices, efficiency is very much look into for a good performance device.

In order to read the efficiency on a time changing scale, power out over power in is used instead.

$$Efficiency = \frac{P_{OUT}}{P_{IN}}$$

An ideal circuit without any loss will yield 100% efficiency. However in reality, losses are factors that are bound to happen. Those factors include, thermal loss, coupling loss, power factor, defect, etc. In terms of electronic devices, most of the loss came from thermal loss as in $P = i^2R$, where i is the current and R is the resistance.

The easiest method of obtaining the power at certain location and time is given as

$$P(t) = VIcos(\omega_1 t + \theta_1)cos(\omega_2 t + \theta_2)$$

where V and I represent voltage and current, and $cos(\omega_1 t + \theta_1)$ and $cos(\omega_2 t + \theta_2)$ represent the AC characteristic of V and I respectively. By tapping on to the current and voltage of the input and output, efficiency can be determined.

2.3.2 Digital Voltage Measuring

In the context of electrical and electronic, voltage and current are the important components that are always put into consideration. It basically tells the amount, force and direction of electron charge is flowing.

There are many way of designing a measuring device. From analogue to digital design of volt and am meter, each with their own advantage and disadvantage. Analogue does not have to rely on external power supply to output value. It rather depends on the electromagnet that is generating through the galvanometer to swing the needle to display output. On the other, hand, digital voltmeter (DVM) basically uses microcontroller feature such as IO pin and ADC to convert into digital value to display on LCD screen. To turn on the microcontroller and LCD, external power source is required.

Below is a 10 Bits ADC conversion taken from PIC24 microcontroller datasheet.

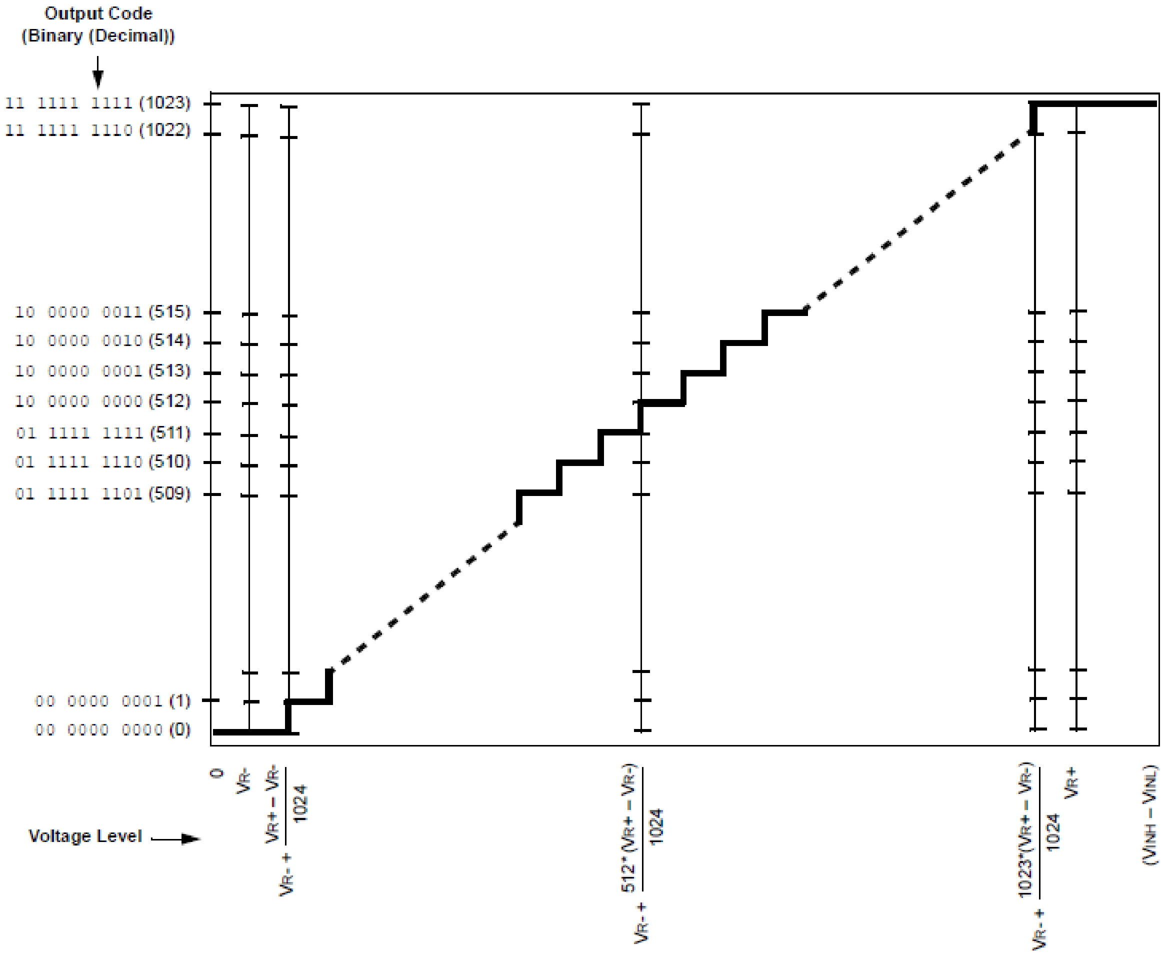


Figure 2.5: ADC Sampling

This conversion basically applies to most of the microcontroller used where the sampling is

$$ADC = \frac{V_{OUT} - V_{(ref-)}}{V_{(ref+)} - V_{(ref-)}} * 1024$$

In addition, microcontroller nowadays also come with more than 1 ADC input and output channel. This allow more flexible used of ADC conversion for varieties of application. However, limitation of microcontroller is bound to have and is shown as in the specification table below.

Table 2.2: ADC Module Specification

TABLE 27-18: ADC MODULE SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Device Supply							
AD01	AVDD	Module VDD Supply	Greater of VDD – 0.3 or 2.0	—	Lesser of VDD + 0.3 or 3.6	V	
AD02	AVSS	Module VSS Supply	VSS – 0.3	—	VSS + 0.3	V	
Reference Inputs							
AD05	VREFH	Reference Voltage High	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	Reference Voltage Low	AVSS	—	AVDD – 1.7	V	
AD07	VREF	Absolute Reference Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
Analog Input							
AD10	VINH-VINL	Full-Scale Input Span	VREFL	—	VREFH	V	(Note 2)
AD11	VIN	Absolute Input Voltage	AVSS – 0.3	—	AVDD + 0.3	V	—
AD12	VINL	Absolute VINL Input Voltage	AVSS – 0.3		AVDD/2	V	
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	2.5K	Ω	10-bit
ADC Accuracy							
AD20b	Nr	Resolution	—	10	—	bits	
AD21b	INL	Integral Nonlinearity	—	±1	<±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22b	DNL	Differential Nonlinearity	—	±1	<±1.25	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23b	GERR	Gain Error	—	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24b	EOFF	Offset Error	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25b	—	Monotonicity[1]	—	—	—	—	Guaranteed

Note 1: The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.
2: Measurements taken with external VREF+ and VREF- used as the ADC voltage reference.

Limitation such as reference voltage, input voltage and etc are limited to the microcontroller maximum specification.

Therefore, to compensate reading for higher voltage level, a modified version of voltage ratio maybe used to overcome those limitations. Topology such as resistive voltage divider (RVD), T-attenuator, Pi-attenuator (Hayt & Kemmerly, 1971) allows calibration and attenuation with maximum power transfer and precision for DVM.

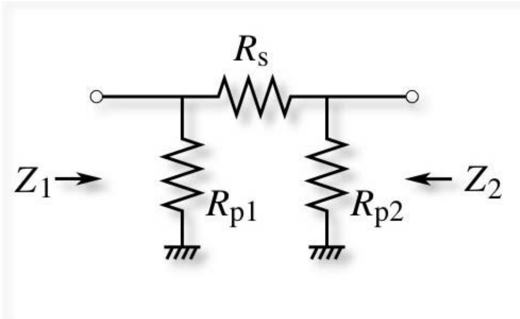


Figure 2.6: Pi Attenuator

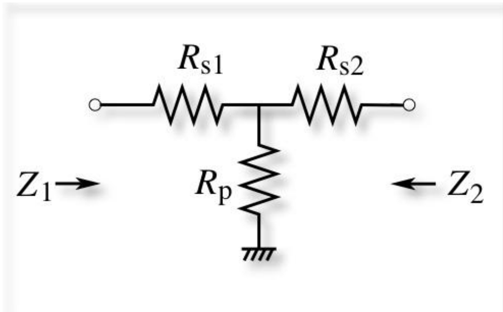


Figure 2.7: T Attentuator

2.3.3 Digital Current Measuring

As for digital ammeter, the complication lies on the integrating circuit that convert current into voltage to feed into ADC of microcontroller. This is due to the fact that ADC used by microcontroller can only accept voltage as input. High current may potentially damage the microcontroller.

The way of connecting the ammeter to the load can either be low side current sensing or high side current sensing. Low side is where the ammeter is connected to the ground side of the load. On the hand, high side is where the ammeter is connected at the path where the power supply is. Below is the diagram illustrates low and high side current sensing.

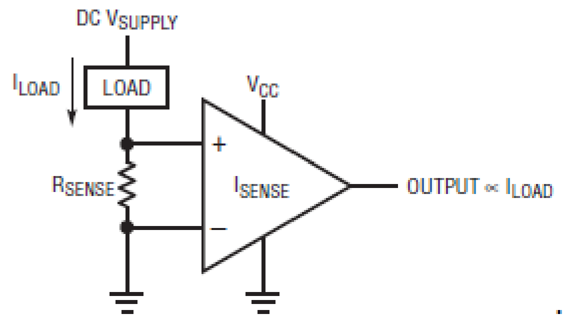


Figure 2.8: Low Side Current Sensing

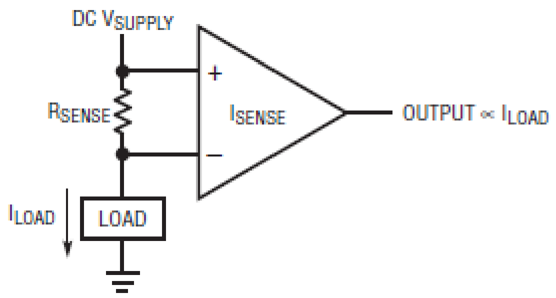


Figure 2.9: High Side Current Sensing

The advantage and disadvantage of each connection is shown as in the table below:

Table 2.3: Advantage and Disadvantage of Current Sensing Mode

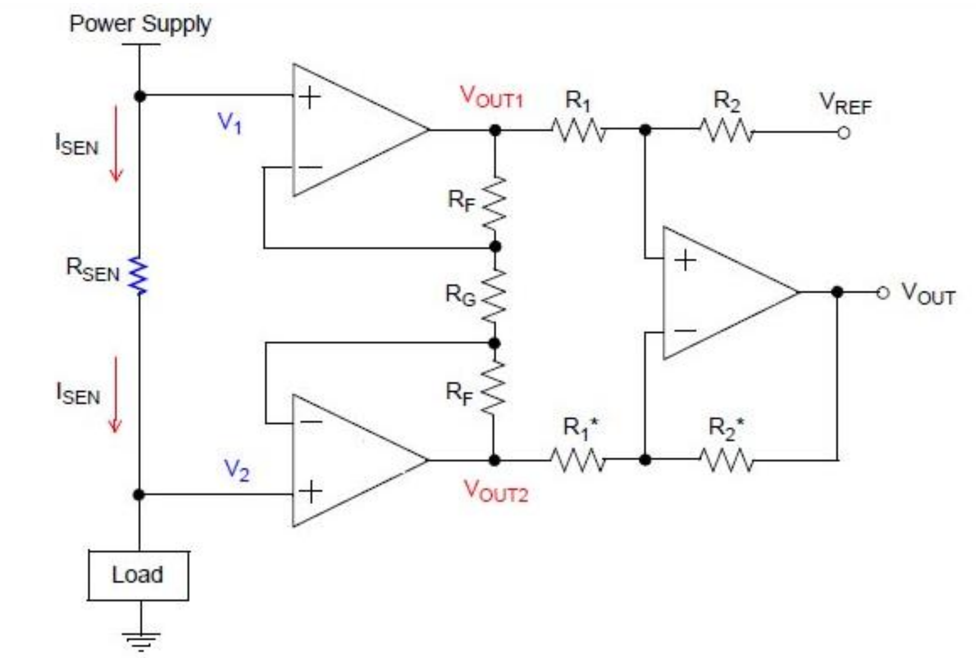
	Advantage	Disadvantage
Low Side Current Sensing	• Low input common mode voltage • Ground referenced output voltage • Easy single supply design	• Load lifted from direct ground connection • Load activated by accidental short at ground end load switch • Short circuiting is not detected
High Side Current Sensing	• Load is grounded • Load not activated by accidental short at power connection • High load current caused by short is detected	• High input common mode voltages (often very high) • Output needs to be level shifted down to system operating voltage levels

(Regan, Munson, Sevastopoulos, Zimmer, & Stokowski, 2005)

2.3.3.1 Current Conversion

There is various way of designing circuit for current sensing circuit. Among the common method is using shunt resistor to act as the voltage tapper for differential circuit. The differential circuit and the shunt resistor is what convert current sensing into voltage for microcontroller A/D converter.

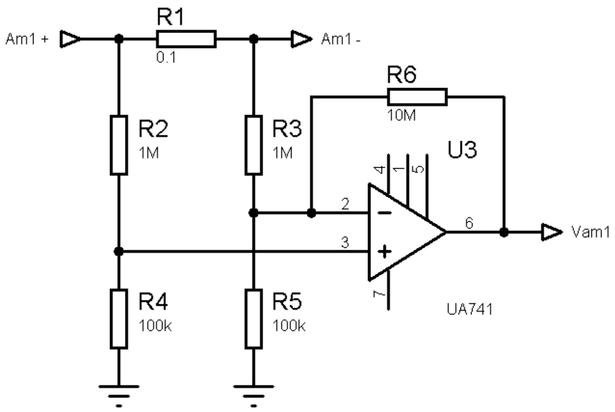
Below is an instrumental amplifier circuit which consist of buffer on the input side and differential amplifier. This kind of circuit is useful for sensing high and low side current sensing. However for high side, it is limited to the operational amplifier common mode voltage (supply voltage for opamp).



(Zhen, 2010)

Figure 2.10: Instrumental Amplifier Circuit

On the other hand, the circuit below is a modified version of a differential circuit which include attenuation on the input. The attenuator basically is a voltage divider which uses to step down the voltage level to an acceptable voltage that is within the common mode voltage.



(Blick, 2005)

Figure 2.11: High Side Current Sensing Differential Circuit

Other than that, there is also various kind of configuration of differential amplifier which can be found by IC manufacturer that is compacted it into single IC. These allow a simpler approach of setting up current converter module.

Precautions for these kinds of current sensing do come with a cost. That is the shunt resistor in some way or another will affect the voltage drop between the supply and the load. Therefore, shunt resistor usually need to be small in value, at the same time have a high tolerance value. This is to avoid deterioration of accuracy when measuring the current flow through it.

2.3.4 Microcontroller Unit

In this literature review, the microcontroller we used will be focusing on PIC24FJ64GA002. The reason of using this is because of the easily available on the market, fast (2 clock cycle per instruction), and sufficient for the system we are building.

Advantage of this include high speed ADC module. Since this is one of the most frequent module is going to be used for this measuring devices. High speed is able to save us the much needed time for processing later on. Besides that, other function such as communication module are available. Those include I2C and UART.

Finally, Real Time Operating System such as uC/OS can be used and easily be programmed into the microcontroller. uC/OS is a preemptive, real-time deterministic multitasking kernel for microprocessors, microcontrollers. It allow up to 250 application task to run simultaneously (multitasking). This allows us to divide task to be run according to priority, at the same time maintaining synchronism at all time for the task set.

2.4 Electric Vehicle

2.4.1 Background

With the increase concern on energy conservation and environmental protection throughout the world, strong interest in electric vehicle (EV) technology is on the rise.

The advantages of EV toward the economy and resources have plenty. They are summarized as below (Chau & Wang, 2005).

- High energy efficiency. Energy conversions to motion for EV and combustion engine are 18% and 13% respectively.
- Energy diversification. Electricity can be generated from all kind of power source such as thermal, solar, wind, hydro and etc.
- Load equalization. In power system, non-stockable energy at non-peak hours is utilized by charging vehicle at night.
- Zero local exhaust emissions. EV emissions are only 2% in carbon monoxide, 76% in carbon dioxide.

Since EV are the already exist product that uses energy storage as a critical component in the system. Understanding further in what kind of technology may also gives a hand or two in producing a better product for this project discussion.

2.4.2 EV Systems and Subsystems

In general, electrical configuration of EVs includes three major subsystems. They are electric propulsion, energy source, and auxiliary (Chau & Wang, 2005).

The electrical propulsion is the subsystem where energy in injected into it to propel the moving motion of the drive. It comprises of electronic controller, power converter, electric motor, mechanical transmission, and driving wheels. On the other hand, the energy source subsystem involves the energy storage, energy management, and recharging of energy. And finally the auxiliary consist of other form of unit which requires energy source such as temperature sensor and steering.

The electrical and mechanical link is shown as in the figure below:

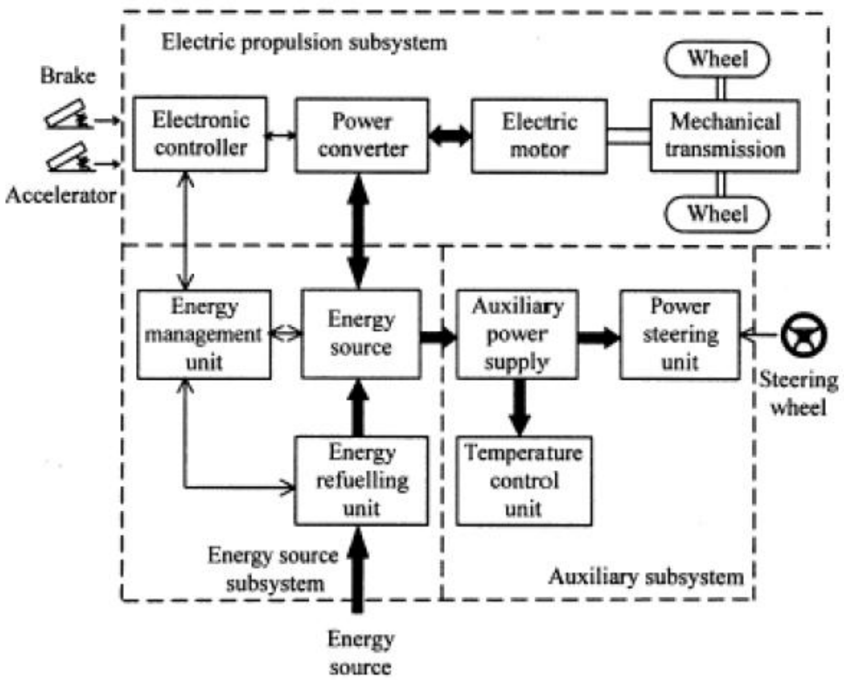


Figure 2.12: General electrical configuration of EVs

2.4.3 Energy Source Comparison

Referring to figure below, a battery is a well-known energy source that is capable of the mass storage of energy; however it has a low power-output density (Baisden & Emadi, 2004). On the other hand, supercapacitor also known as ultra capacitor for the table below has little storage; however it can supply a large burst of power. Hence, it is common for industry maker to harness both together to meet the demand of driving the electric motor (Baisden & Emadi, 2004).

Table 2.4: Battery versus Ultra-Capacitor Performances

Performance	Battery	Ultra-capacitor
Specific energy (storage)	10-100 $W-h/Kg$	5-10 $W-h/Kg$
Specific power (delivery)	<1000 W/Kg	<10,000 W/Kg
Charge/discharge efficiency	50-85%	85-98%
Life expectancy	3 years	10 years

(Baisden & Emadi, 2004)

Alternative comparison with combustion engine energy sources are shown as below:

Table 2.5: Natural Source Energy Specification

Energy Source	**Nominal Specific Energy (Wh/kg)**
Gasoline	12,500
Natural gas	9350
Methanol	6050
Hydrogen	33,000
Lead-acid battery	50
Lithium-polymer battery	200
Ultra-Capacitor	10

(Husain, 2003).

As shown from both the table, there is a clear indication that in terms of energy density, electrical energy storage is still far off from the combustion energy storing counterpart.

Subsequently, the outcome of this case study doesn't actually show the progress of supercapacitor acting as energy storage that is large enough to power an electric vehicle. From the specification above, the potential of using supercapacitor is in fact possible if and only if large quantity of supercapacitor is used together for the large energy to be stored. However, until now the best source of energy for this high power machine is still the hybrid by combining battery and capacitor to get the better output.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, we will discuss on the methodology used to complete the project. This includes the boost converter circuit as well as measuring device for the boost converter efficiency. By using most of the fundamental found from the literature review, the best suitable design is implemented as the finishing product for the final year project. In the end, a thorough analysis is performed to obtain a characteristic for the design boost converter.

Before everything that can be made, a standard for our device is needed. This is specified as the specification and requirement for the whole designing process. They are:

- Input Voltage < 5.4V
- Output Voltage = 12V
- Maximum Output Current = 1A

3.2 Job Division

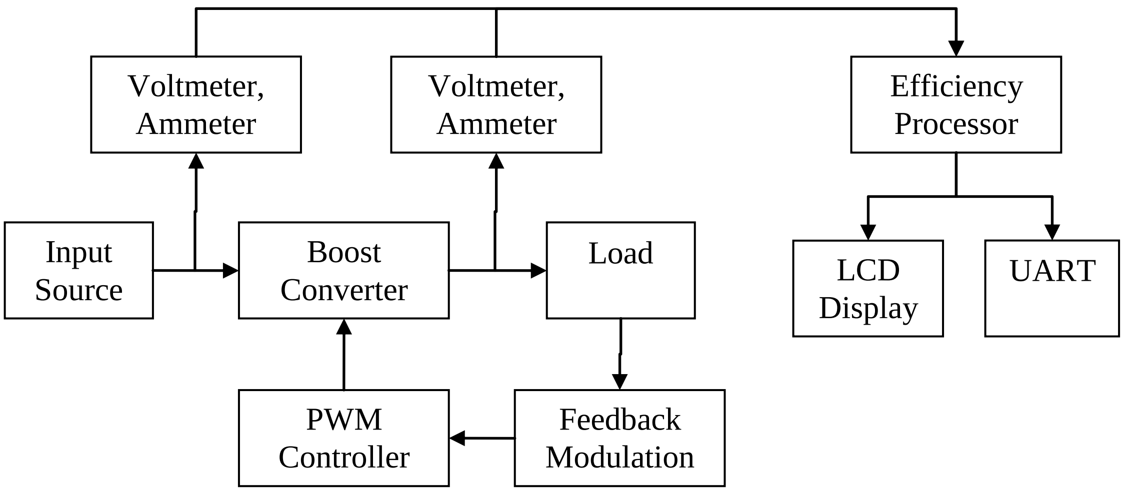
To begin, the hardware design of this project is break into 2 sections. They are Converter Design and Efficiency Measuring Device. From there, we further break this section into sub-components to allow detail design and analysis for it.

This project is a 2 member's project. Hence, the division of work for the methodology is break accordingly to the section separated earlier.

Table 3.1: Job Division

Person In charge	**Job Division**	**Sub-Component**
Foo Shin Loong	Converter Design	Boost Converter
		PWM Controller
		Feedback Modulation
Choong Pak Wai	Measuring Device	Voltmeter
		Ammeter
		Efficiency Processor
		Data transfer Module

The whole system is link as in the diagram below.



3.3 Efficiency Measuring Device

3.3.1 Microcontroller

The implementation of Analogue to Digital Converter (ADC) for voltmeter and ammeter, computation, LCD controller and Universal Asynchronous Receiver Transmitter (UART) will be based on a main microcontroller. There are many microcontrollers that are well suit for the application of our system, and we have come down to selecting PIC24FJ64GA002 as the brain for our system.

The requirement for this device requires minimum 4 ADC input pin, digital I/O, and communication module. PIC24FJ64GA002 is a 28 pin microcontroller that come with 21 I/O pin. 13 ADC input port of 500 ksps of conversion rate that are integrated with the I/O pin. Finally, the mode of transferring sampled data allows UART, I2C and SPI. Above it all, this is a DIP package which is easy to be used for hardware implementation.

The minimum configuration for is microcontroller is set as below schematic

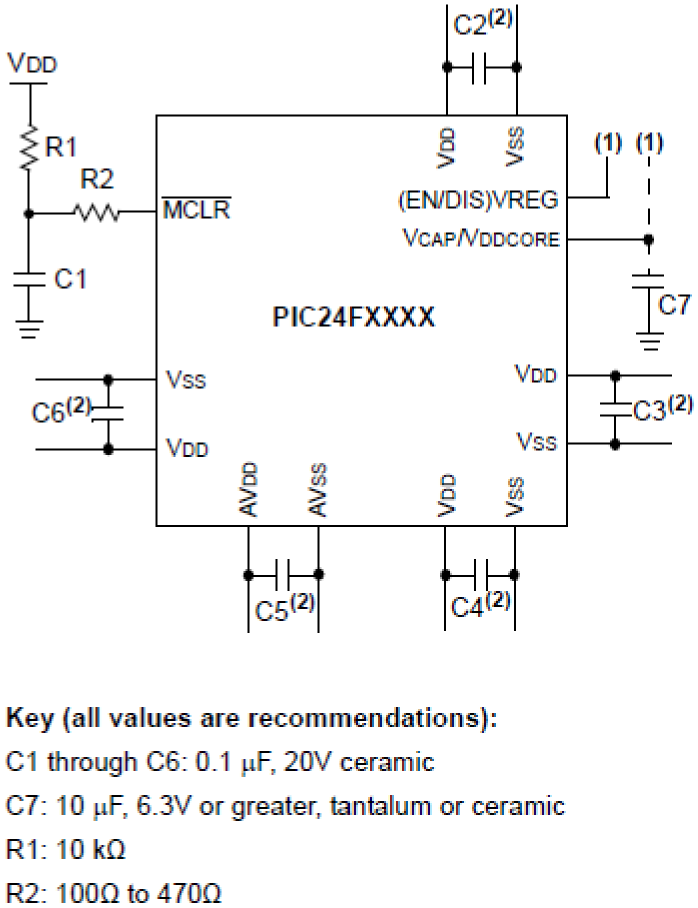


Figure 3.1: Microcontroller Basic Configuration

3.4 Power Sensing Device

3.4.1 Voltage Measurement

The converter we build using 2 series capacitor allow maximum voltage level at the input to be at 5.4V. Meanwhile, the boost output is at 12V. Both the voltage level is above the typical threshold of the microcontroller reference voltage. To make life easier for programming, such as implementing multiplexing, we set the reference voltage to be fixed at 3.3V. This is equivalent to the supply voltage for the microcontroller.

Next, we step down the input and output voltage to a measurable level, an attenuator is designed using basic voltage divider and buffer.

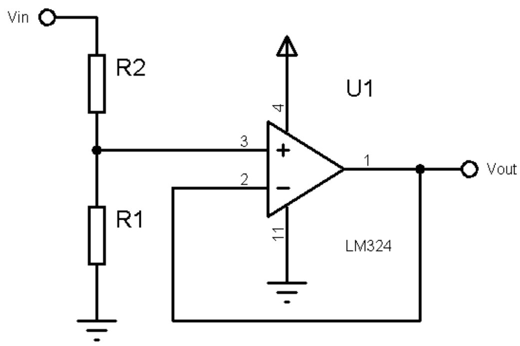


Figure 3.2: Attenuation Using Voltage Divider

The ratio attenuate is later multiply back through programming to display the actual value.

3.4.1.1 Input Voltage Divider

Assuming the input voltage ratio we used is 2. We need to reduce the voltage level by half. Thus, R_1 and R_2 value are obtain as below.

$$ratio = \frac{V_{IN}}{V_{ref}} = \frac{5.4}{3.3} = 1.6$$

Set R_1 = 1 kΩ

$$ratio = \frac{R_1}{R_1 + R_2}$$

$$R_2 = 1k\ \Omega$$

3.4.1.2 Output Voltage Divider

Assuming the input voltage ratio we used is 8. We need to reduce the voltage level by half. Thus, R_1 and R_2 value are obtain as below.

$$ratio = \frac{V_{OUT(\max)}}{V_{ref}} = \frac{24}{3.3} = 7.2$$

Set R_1 = 1 kΩ

$$ratio = \frac{R_1}{R_1 + R_2}$$

$$R_2 = 7k\ \Omega\ \approx 6.8\ k\Omega\ (practical\ value)$$

3.4.2 Current Measurement

As for ammeter, there aren't any voltage levels that may used to be sample through the ADC. This is where current sensing circuit is used to convert current to voltage. The constant ratio now is in term of ampere per volt (A/V). Current sensing device usually require a low resistive component connected in a shunt manner. By comparing voltage different between the 2 ends of the shunt resistor, we can estimate the current flow.

Since the focus of the design is on a low side current measurement, differential amplifier is not needed anymore. The differential amplifier is simplified into a non-inverting amplifier and inverting amplifier, since one side of the input is always connected to ground.

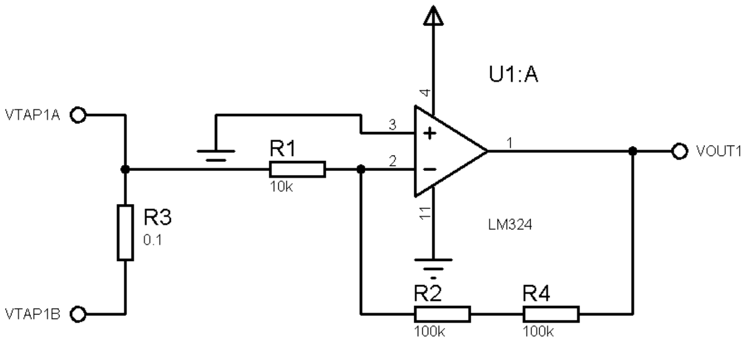


Figure 3.3: Rsense using Inverting Amplifier

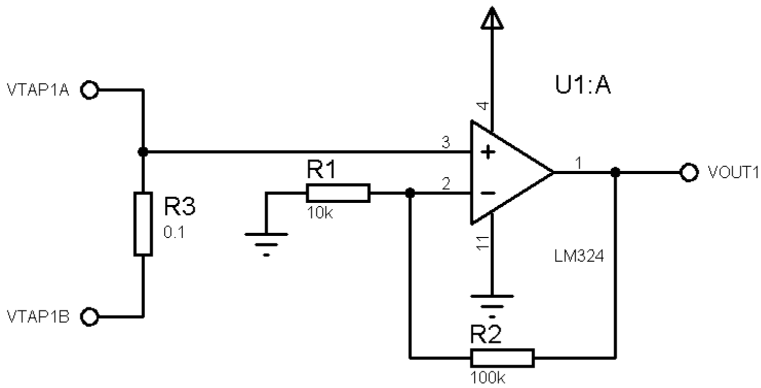


Figure 3.4: Rsense using Non-Inverting Amplifier

3.4.2.1 Input Current Conversion

An inverting amplifier is used because current is expected to flow from the ground out. Hence value read through the shunt resistor is a negative value.

$$V_{OUT} = -\frac{R_2}{R_1}\left(-I_{IN}R_{SHUNT}\right)$$

Base on the design of the boost converter, the input current is expected to reach maximum of 5A. Simultaneously, the maximum input voltage for microcontroller ADC is 3.3V. Hence, at maximum conversion 5A is converted into 3V. Setting those as the parameter, R1 and R2 is determined.

Set R1 = 10kΩ

$$R_2 = \frac{V_{OUT} R_1}{I_{IN} R_{SHUNT}} = \frac{(3V)(10k\Omega)}{(5A)(0.1\Omega)} = 60\ k\Omega$$

3.4.2.2 Output Current Conversion

A non-inverting amplifier is used because current is expected to flow from the load to the ground. Hence value read through the shunt resistor is a positive value.

$$V_{OUT} = \left(\frac{R_2}{R_1} + 1\right)(-I_{IN} R_{SHUNT})$$

Base on the design of the boost converter, the output current is expected to reach maximum of 2A. Simultaneously, the maximum input voltage for microcontroller ADC is 3.3V. Hence, at maximum conversion 2A is converted into 3V. Setting those as the parameter, R1 and R2 is determined.

Set R1 = 10kΩ

$$R_2 = \left(\frac{V_{OUT}}{I_{IN} R_{SHUNT}} - 1\right) R_1 = \left(\frac{(3V)}{(2A)(0.1\Omega)} - 1\right)(10k\Omega) = 140\ k\Omega$$

3.5 ADC sampling

To take advantage of the high ADC sampling for PIC24FJ64GA002 of up to 500k samples/sec, a carefully structure coding which is controlled by RTOS allows this to be done more effectively.

Assuming the sample rate is 500 kHz, total time per sample is

$$T = \frac{1}{f} = \frac{1}{500\ kHz} = 2\ us$$

Even with the multiple sampling to obtain average value which is discuss in the Chapter 4, the total time sampling still does not affect the cycling of serving each individual task. For ADC task, the serving of the task is done at every 200ms (discuss in the later topic). Comparing 200ms and multiple of 2us (using averaging of 50 samples per cycle), this still does not affect much toward the overall performance of servicing time.

3.6 Data Display and Transfer

The device designed has 2 mode of data display. They are the LCD display and UART. The LCD used is JHD 162A which is able to display 2 rows of 16 characters display. Due to limit number of pins available on PIC24 that we are using, a 4-bits data transferring method is used for programming the LCD.

LCD interface to the microcontroller as shown in the schematic below.

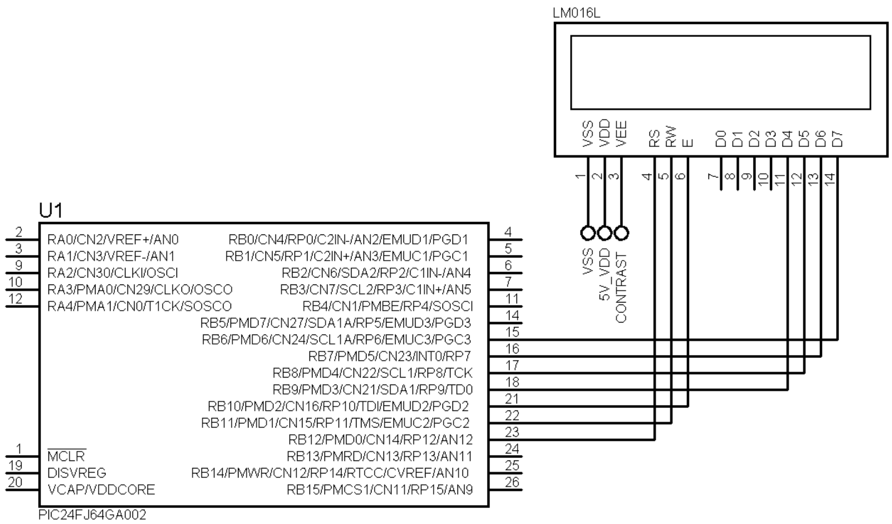


Figure 3.5: 4 Bits LCD Module Interconnection

LCD allows easy access of seeing real time simulation result. And the full program for LCD library is available on the appendix

On the other hand, running simultaneously with LCD display, the data is transmitted through UART to be collected via terminal port computer. This is to allow us to have a thorough analysis of what is tested in table form. Doing so, we can even inspect breakthrough found during simulation.

For PIC24 that we are using, there are 2 UART modules available. However, we only use 1 which is enough for the need of data transferring to computer. Before anything can be done to the UART module, baud rate need to be set. We have chosen 19200 as our baud rate as it is moderate between high speed and error frequency.

The connection of the circuit is a direct pin connection through a PICKIT2 as a RS232 intermediate.

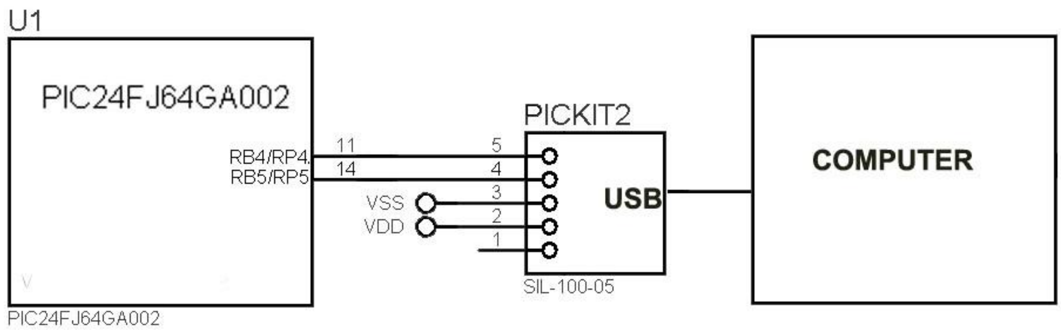


Figure 3.6: UART Module Interconnection

Below is a series of screen shot of how data collected from UART is transferred to excel form for analysis. The data that is send to UART are input current, input voltage, output current, output voltage and system efficiency.

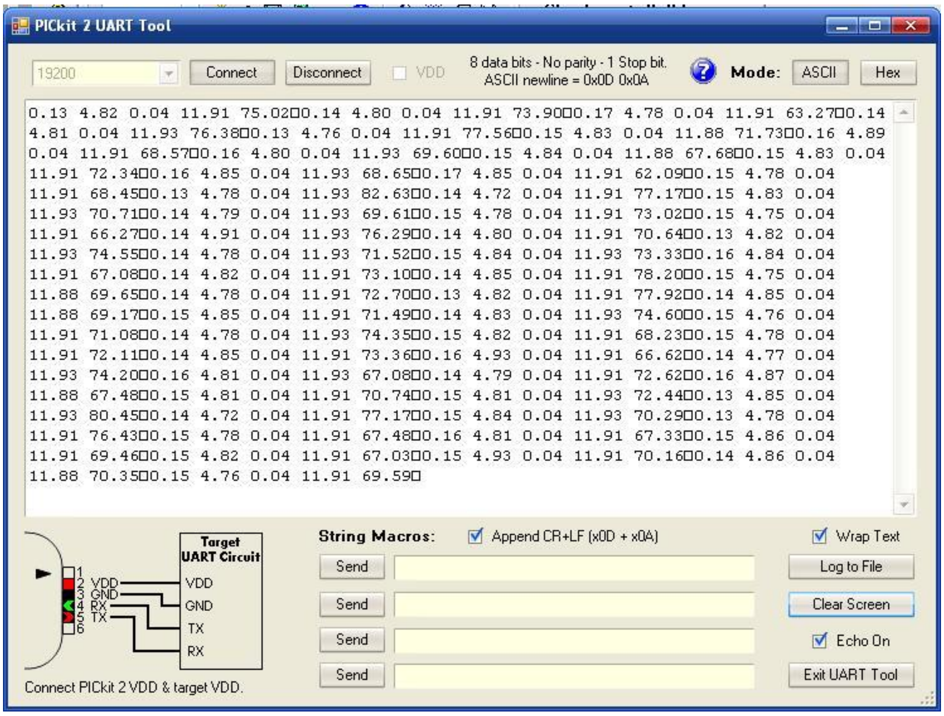


Figure 3.7:UART tools provided from PICkit 2 software

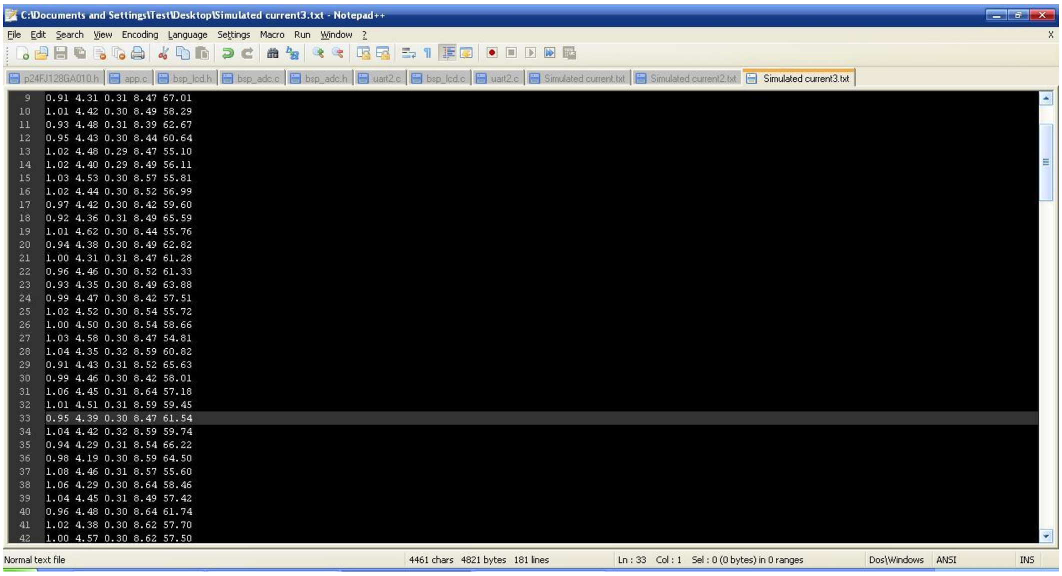
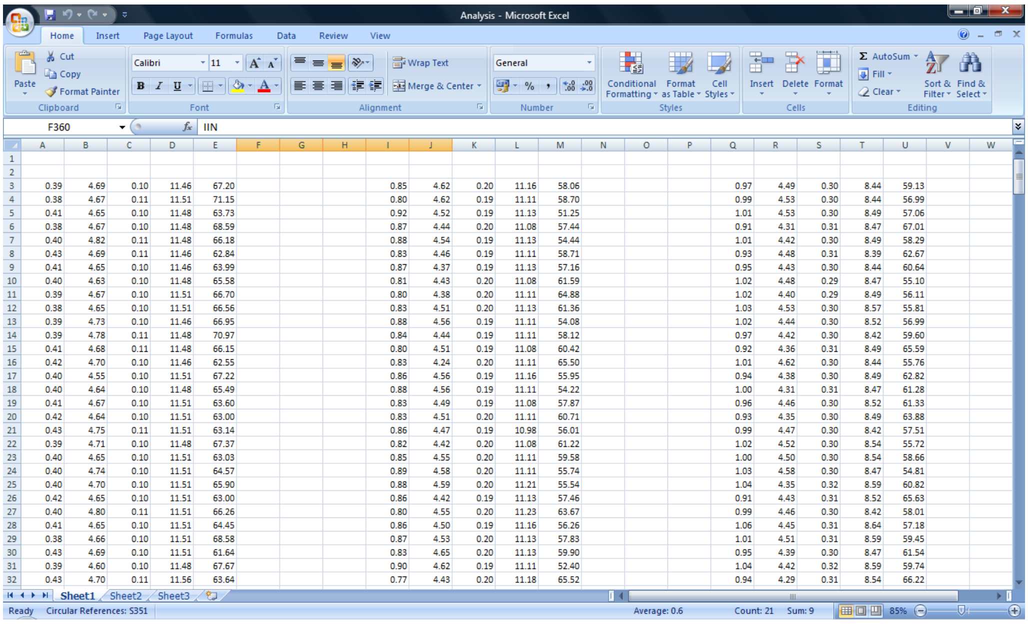


Figure 3.8: Log File copied from UART Tools



	A	B	C	D	E	I	J	K	L	M	Q	R	S	T	U
3	0.39	4.69	0.10	11.46	67.20	0.85	4.62	0.20	11.16	58.06	0.97	4.49	0.30	8.44	59.13
4	0.38	4.67	0.11	11.51	71.15	0.80	4.62	0.19	11.11	58.70	0.99	4.53	0.30	8.44	56.99
5	0.41	4.65	0.10	11.48	63.73	0.92	4.52	0.19	11.13	51.25	1.01	4.53	0.30	8.49	57.06
6	0.38	4.67	0.10	11.48	68.59	0.87	4.44	0.20	11.08	57.44	0.91	4.31	0.31	8.47	67.01
7	0.40	4.82	0.11	11.48	66.18	0.88	4.54	0.19	11.13	54.44	1.01	4.42	0.30	8.49	58.29
8	0.43	4.69	0.11	11.46	62.84	0.83	4.46	0.19	11.11	58.71	0.93	4.48	0.31	8.39	62.67
9	0.41	4.65	0.10	11.46	63.99	0.87	4.37	0.19	11.13	57.16	0.95	4.43	0.30	8.44	60.64
10	0.40	4.63	0.10	11.48	65.58	0.81	4.43	0.20	11.08	61.59	1.02	4.48	0.29	8.47	55.10
11	0.39	4.67	0.10	11.51	66.70	0.80	4.38	0.20	11.11	64.88	1.02	4.40	0.29	8.49	56.11
12	0.38	4.65	0.10	11.51	66.56	0.83	4.51	0.20	11.13	61.36	1.03	4.53	0.30	8.57	55.81
13	0.39	4.73	0.10	11.46	66.95	0.88	4.56	0.19	11.11	54.08	1.02	4.44	0.30	8.52	56.99
14	0.39	4.78	0.11	11.48	70.97	0.84	4.44	0.19	11.11	58.12	0.97	4.42	0.30	8.42	59.60
15	0.41	4.68	0.11	11.48	66.15	0.80	4.51	0.19	11.08	60.42	0.92	4.36	0.31	8.49	65.59
16	0.42	4.70	0.10	11.46	62.55	0.83	4.24	0.20	11.11	65.50	1.01	4.62	0.30	8.44	55.76
17	0.40	4.55	0.10	11.51	67.22	0.86	4.56	0.19	11.16	55.95	0.94	4.38	0.30	8.49	62.82
18	0.40	4.64	0.10	11.48	65.49	0.88	4.56	0.19	11.11	54.22	1.00	4.31	0.31	8.47	61.28
19	0.41	4.67	0.10	11.51	63.60	0.83	4.49	0.19	11.08	57.87	0.96	4.46	0.30	8.52	61.33
20	0.42	4.64	0.10	11.51	63.00	0.83	4.51	0.20	11.11	60.71	0.93	4.35	0.30	8.49	63.88
21	0.43	4.75	0.11	11.51	63.14	0.86	4.47	0.19	10.98	56.01	0.99	4.47	0.30	8.42	57.51
22	0.39	4.71	0.10	11.48	67.37	0.82	4.42	0.20	11.08	61.22	1.02	4.52	0.30	8.54	55.72
23	0.40	4.65	0.10	11.51	63.03	0.85	4.55	0.20	11.11	59.58	1.00	4.50	0.30	8.54	58.66
24	0.40	4.74	0.10	11.51	64.57	0.89	4.58	0.20	11.11	55.74	1.03	4.58	0.30	8.47	54.81
25	0.40	4.70	0.10	11.51	65.90	0.88	4.59	0.20	11.21	55.54	1.04	4.35	0.32	8.59	60.82
26	0.42	4.65	0.10	11.51	63.00	0.86	4.42	0.19	11.13	57.46	0.91	4.43	0.31	8.52	65.63
27	0.40	4.80	0.11	11.51	66.26	0.80	4.55	0.20	11.23	63.67	0.99	4.46	0.30	8.42	58.01
28	0.41	4.65	0.10	11.51	64.45	0.86	4.50	0.19	11.16	56.26	1.06	4.45	0.31	8.64	57.18
29	0.38	4.66	0.10	11.51	68.58	0.87	4.53	0.20	11.13	57.83	1.01	4.51	0.31	8.59	59.45
30	0.43	4.69	0.10	11.51	61.64	0.83	4.65	0.20	11.13	59.90	0.95	4.39	0.30	8.47	61.54
31	0.39	4.60	0.10	11.48	67.67	0.90	4.62	0.19	11.11	52.40	1.04	4.42	0.32	8.59	59.74
32	0.43	4.70	0.11	11.56	63.64	0.77	4.43	0.20	11.18	65.52	0.94	4.29	0.31	8.54	66.22

Figure 3.9: Data is converted into excel format using the import wizard

Full program for the UART module is available in the appendix.

3.7 Program Code Architecture

The code is what links every component discussed earlier to work as a unit. Since there are so many modules to be considered (LCD, UART, ADC), Real Time Operating System (RTOS) is used. Doing so, we can set context switch repetitively changing from module to module in a systematically manner according to priority of individual task.

Generally, we have broken this system into 3 main tasks. They are getting the ADC sample, LCD display, and UART data transmission. The Priority of each task is set as below:

1. Getting ADC Sample
2. UART Data Transmission
3. LCD Display

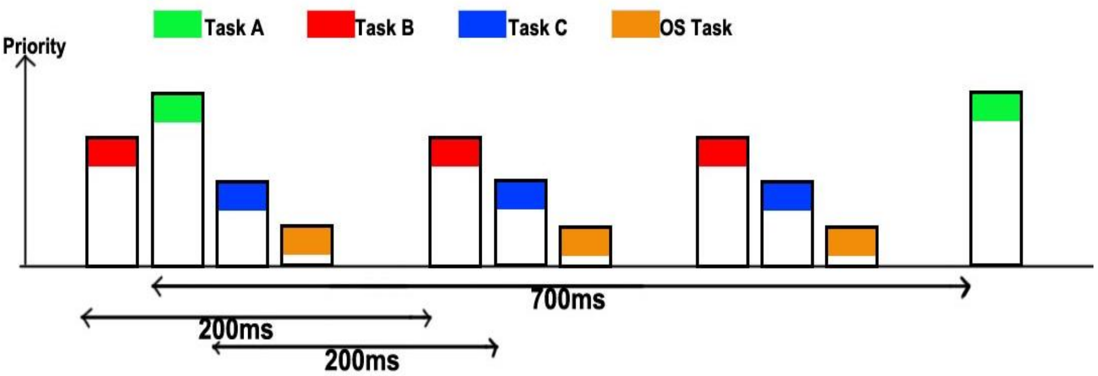
By determine the priority the urgency of servicing each task and be adjusted using OSTimeDly() function. Prior to the time elapsing of each task is being done in a systematically order. RTOS also provide a much simpler code writing as well as using readymade modules through Board Support Packages (bsp). Doing so, porting the code to a higher specification PIC is also made easier in the future.

In order to avoid shared data problem due to sharing of variable for input/output current and voltage, semaphore is used as in below.

- Get semaphore - OSSemPend(DispSem, 0, &err);
- Release semaphore - OSSemPost(DispSem);

A semaphore is a key that your code acquires in order to continue execution. If the semaphore is already in use, the requesting task is suspended until the semaphore is released by its current owner.

And the flow of the routine is shown in the flow chart below.



The priority level of each task is set as below.

```
#define   ADC_TASK_PRIO                    5   //Task B
#define   UART2_TASK_PRIO                  6   //Task C
#define   LCD_TASK_PRIO                    4   //Task A
#define   OS_TASK_TMR_PRIO                 10 //OS Task
```

Since LCD display is only used as an indication and rough reading, the delay is set to be the longest. On the other hand, ADC (Task B) and UART (Task C) is set to be operating one after another, at the interval of 200ms. Doing so, graph plotted in Chapter 4 can be set to interval of 200ms. The rest of the time is controlled by RTOS (OS Task).

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Digital Voltmeter (DVM)

From the study of PIC 24 microcontroller, the design of the DVM is not a perfectly accurate measuring device. Among the issue face when designing this are loading effect when connecting load to the ADC pin. Besides that, fluctuation of sampled ADC also fluctuates at a large scale.

4.1.1 Loading Effect

Below is the ADC pin, input model taken from the datasheet for PIC24 microcontroller that we are using. It is advice that the input resistance is fixed to be less than 5kΩ.

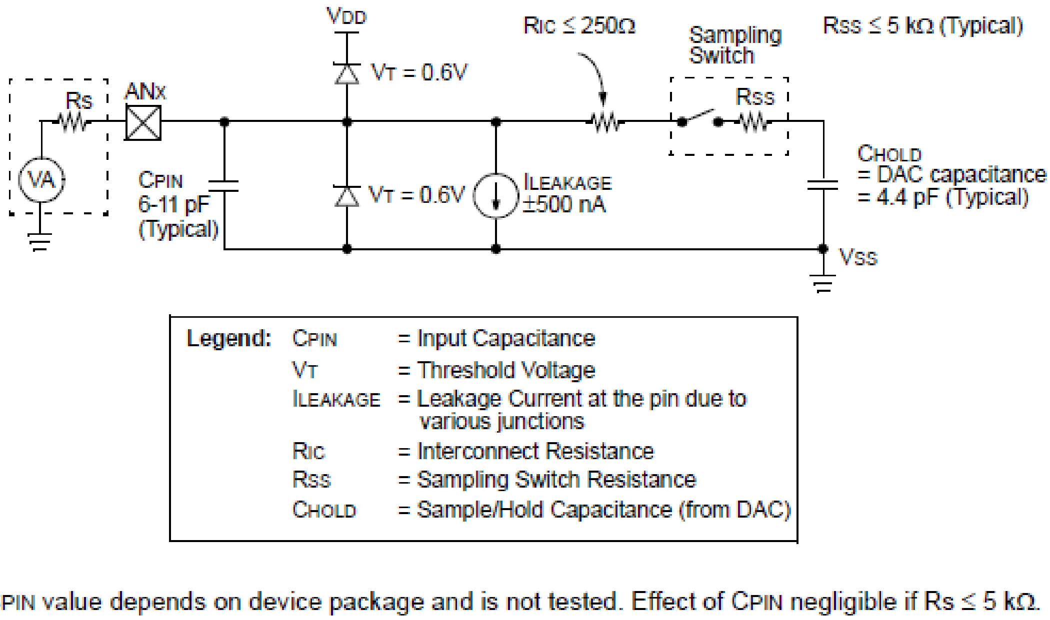


Figure 4.1: Internal ADC Structure

However, when low impedance is used, current flow through it will be increased. By estimation, current flow though it will be:

$$I = \frac{V}{R} = \frac{12}{5k} = 2.4mA$$

Therefore, any value above 5kΩ is sufficient to limit the amount of current flow in to ADC for sampling. In our design, we used value in the range of 10 kΩ. Now, the problem lies on the loading effect that prevents us from measuring value accurately. Therefore, a buffer is added as in diagram below:

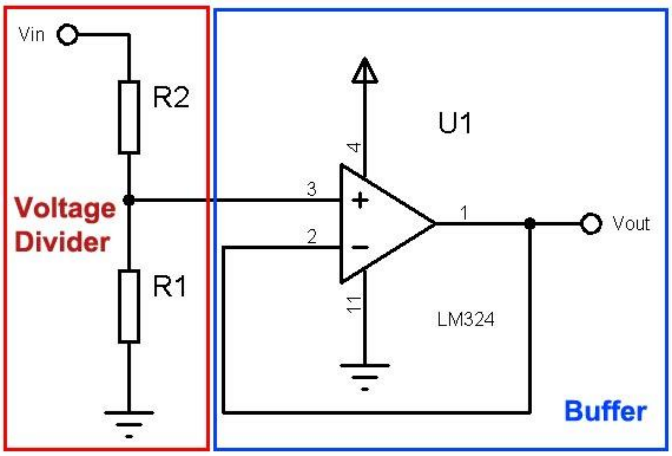


Figure 4.2: Loading Effect Solution

The buffer job is to isolates the attenuator loading effect from the ADC pin by providing a high impedance on the input and low impedance on the output.

4.1.2 Fluctuation of Sampled Value

However, this still doesn’t solve the issue where the sampled value fluctuates. Therefore, the approach taken is using multiple sampling, and taking the average value as the output as below.

```
ADC_sel_channel(ADC_SEL_V0);
for(i = 0; i<20; i++)
{
      readValue += ADC_read();
}
readValue = readValue/20;
```

Since this is a high speed ADC converter, the averaging method used will not affect much toward the whole processing time. Considering the each sample uses 10 times of conversion and taking the average. The overall sampling time is still operating at 50 kHz which is still relatively fast.

The cause of the fluctuation is likely causes by the internal configuration of mircrocontroller such as ADC structure inside the microcontroller. As found from Application Note 800 from Maxim-IC (Maxim-IC, 2002), we notice that high speed application normally require clean clock signal, so that it does not contribute undesired noise to the overal dynamic performance of the system. As in such case, we uses internal generated clock instead of external clock source for sampling. This might be the source that contributes to the jitter produced.

Jitter may be caused by several attribute. They are Thermal Noise and Phase Noise.

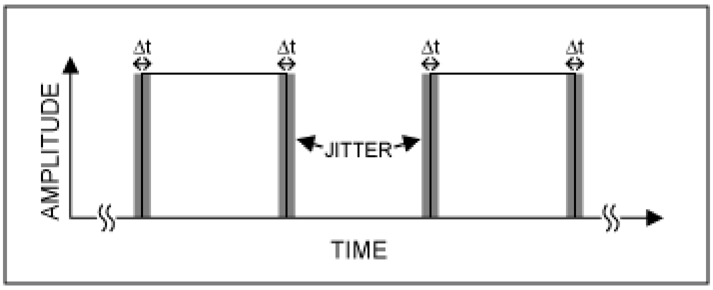


Figure 4.3: Jitter in clock signal that degrades the ADC signal-to-noise ratio.

Though it is mean for high speed devices, chances are this jitteriness may also be the source of faultiness in our device. Jitter generated by a clock source can cause the ADC's internal circuitry to falsely trigger the sampling time, Δt. As a result, we get a false sampling for the analogue input value, ΔA. A clearer picture is observed in Figure 4.4. Note that, sampling capacitance for this PIC24 (C_{HOLD}) is in the range of pF which result is faster charging and discharging time constant. And hence is very sensitive to the slightest clock change.

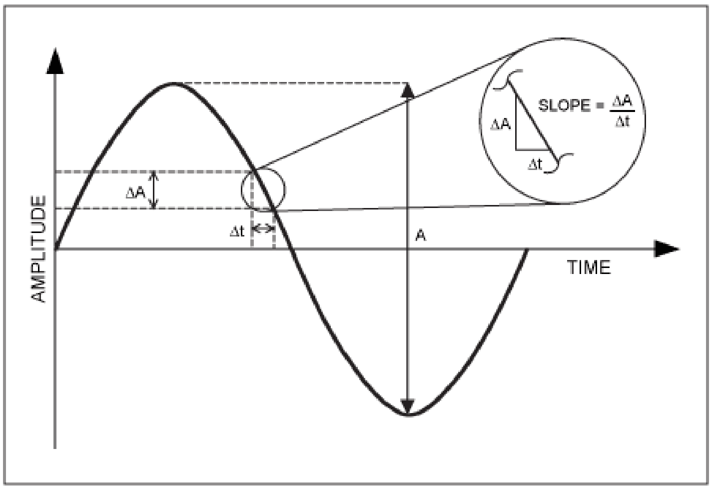


Figure 4.4: Ampliture change causes by jitter

4.2 Accuracy of DVM

The sampling resolution of our device can achieve 0.01V for output voltage and 0.005V for the input voltage.

Resolution of Output Voltage: $\frac{12}{1024} = 0.01\ V/ADC_{binary}$

Resolution of Input Voltage: $\frac{5.4}{1024} = 0.005\ V/ADC_{binary}$

Due to the fluctuation of value as mention in the above topic, the accuracy of this measuring device has some tolerance to cope. Through the measurement of the device against fix voltage, a commercial multimeter UNI-T UT33B is used as the reference for actual value. The sampled value is taken for a short duration using UART is summarized as below.

Table 4.1: Voltmeter analysis for input side

Vin	Measured			Error
Actual	Min	max	ave	
0.00	0.00	0.01	0.00	0.0000
0.30	0.30	0.30	0.30	0.0000
0.50	0.50	0.50	0.50	0.0000
1.00	0.99	0.99	0.99	0.0100
3.00	2.96	2.98	2.97	0.0100
5.00	4.91	5.00	4.95	0.0096
5.40	5.34	5.34	5.34	0.0111
5.60	5.53	5.54	5.54	0.0107
		Average Error		0.0064

Table 4.2: Voltemeter analysis for output voltage

Vout	Measured			Error
Actual	Min	max	ave	
0.00	0.05	0.05	0.05	0.0000
0.36	0.40	0.40	0.40	0.1111
0.50	0.52	0.55	0.54	0.0800
1.00	1.03	1.05	1.05	0.0470
3.00	3.01	3.04	3.04	0.0133
7.00	6.98	7.01	6.98	0.0029
10.00	9.97	9.97	9.97	0.0030
11.00	10.95	10.98	10.95	0.0045
12.00	11.96	11.99	11.97	0.0025
15.00	14.95	14.98	14.96	0.0027
20.00	19.90	19.93	19.92	0.0040
		Average Error		0.0246

The computed percentage of error average for all voltage level is 0.64% and 2.46% for input and output respectively.

The error value is computed using the following equation:

$$Error = \frac{|Measured - Actual|}{Actual}$$

In this analysis, we are trying to determine the consistency of this voltmeter we have designed. Generally speaking, the percentage of error we obtain is relatively low by using multimeter UNI-T UT33B as comparison for the actual value being measured.

We do note that for input voltage, the accuracy tend to be higher at lower voltage. On the other hand, Vout measured have the tendency to be more accurate for the higher measured value. The cause of this variation is likely the cause of multiple effect such as the fluctuation of jitter effect, low performance buffer IC used LM324, and tolerance of the resistor value. Most of the fluctuation by jitter effect has been solve using averaging sampling method. As for the other 2 flaw, this is the physical flaw which only can be overcome by buying higher specification components.

4.3 Digital Am Meter (DAM)

4.3.1 Boost Converter Input Side

To measure the current for the boost converter, there are few factors we need to consider. For instance, the measurement on the high side current sensing is not design properly, loading effect will occurs and this will cause the actual reading of the capacitor and load to be inaccurate. From the experiment done, previously, we notice that the different in the result measured is as high as 20% different from the actual value.

Therefore, a better and more stable measuring solution is done on the low side current sensing. There is slight difference on input and output side of current measurement. We notice that the boost converter is in fact operating in a DCM mode. Due to the low frequency produce from the 555 timer for the PWM, the operating basically changed from CCM to DCM that we actually designed for. Never the less, this is still able to detect by our inverting current converter as shown in the figure below.

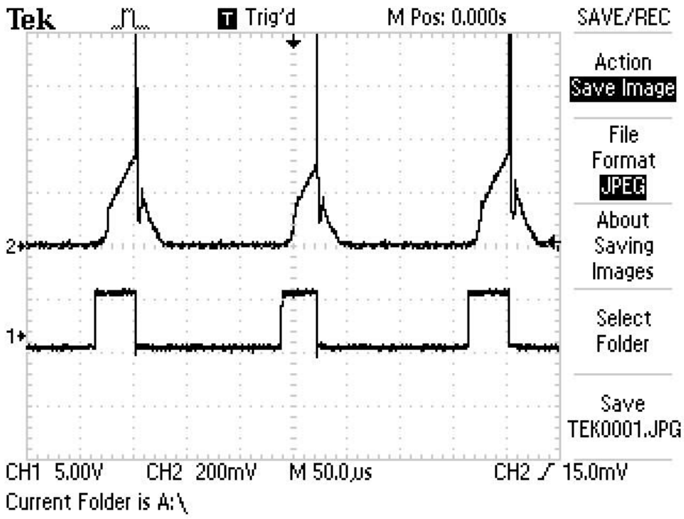


Figure 4.5: Input current at low PWM duty ratio

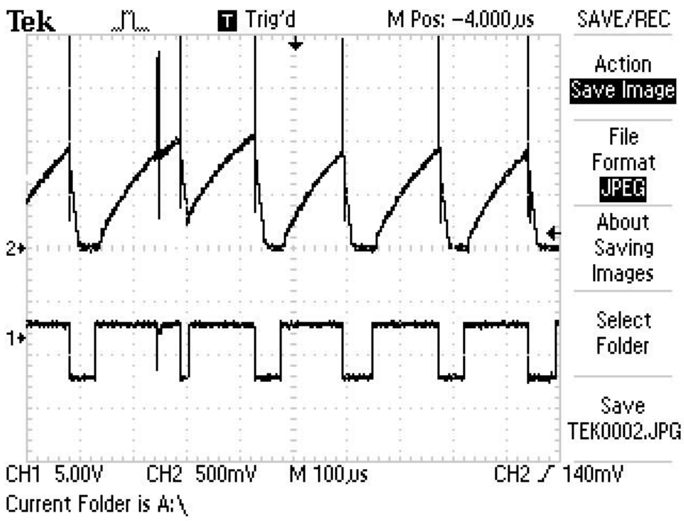


Figure 4.6: Input current at high PWM duty ratio

In Figure 4.5 and 4.6, channel 1 is connected to the PWM at the MOSFET for the switch, and channel 2 is connected to the output of the amplifier.

Note, that input current behave as an alternating current. Therefore, to counter the problem arise, software modification is done on the ADC sampling, so that AC reading is taken instead. Doing so, we can estimate the efficiency of the converter for effectively.

So, for the program part, multiple samples on AC value is taken throughout 1 period. Since the AC frequency is the same as PWM frequency, we can calculate the duration of sampling time as below.

$$T = \frac{1}{f} = \frac{1}{5kHz} = 0.2\ ms$$

Assuming ADC sampling rate is 500 kHz, the amount of sample required is 100 times.

However, that is not the case of our design in hardware, since there are a lot more uncertainty factor that is not included which can cause the speed of sampling rate. Therefore, in order to adjust this to the best value is merely depends on trial and error of programming.

The analysis of sampled data taken from UART for the input side is obtained and summarize as shown in below:

Table 4.3: Input Current Data Sampled Analysis

Input Current, I_{IN}				
Actual	Min	Max	Average	Standard Deviation
0.42	0.35	0.45	0.40	0.02
0.87	0.77	0.92	0.84	0.03
1.02	0.84	1.08	0.94	0.05

The actual reading is taken from the “Topward DC power supply 6303D”. However, this is only used to be considered as the reference value since the actual current flowing through it is alternating as shown in figure 4.5 and 4.6.

Note that the standard deviation for all 3 variable of reading is in the range of 0.02 to 0.05. This highlight the varying or difference of the sampled value compared to the average sampled value to be near to each other and is reliable enough to prove the accuracy of input current. Also, from the figure 4.5 and 4.6 we note that the current is AC. Therefore, it is understandable with the min and max value that varies.

4.3.2 Boost Converter Output Side

Due to the triggering of the MOSFET, by manually looking at the current on the output through a current converter of the non-inverting amplifier, we notice that the load also behave as in pulsating mode.

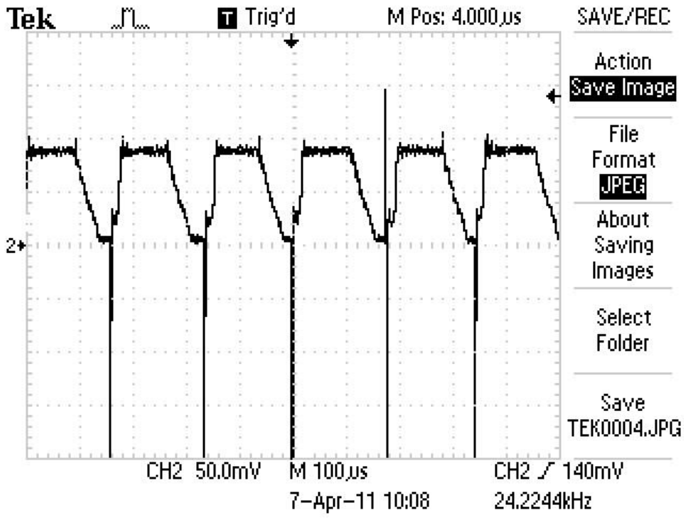


Figure 4.7: Output Current low PWM duty ratio

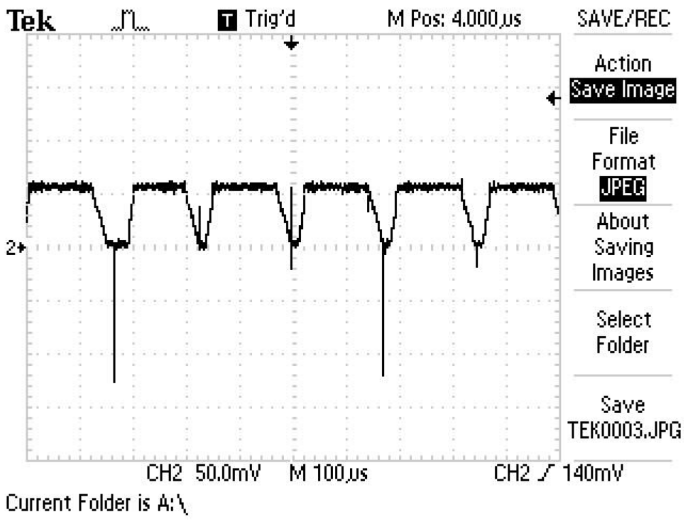


Figure 4.8: Output Current high PWM duty ratio

Therefore, the solution of this is the same as discussed in the previous section for the input side for the current.

The sampled of output current for analysis is summarized in the table below:

I_{OUT}				
Actual	Min	Max	Average	Standard Deviation
0.11	0.10	0.11	0.10	0.00
0.21	0.19	0.20	0.20	0.00
0.33	0.29	0.32	0.30	0.01

The load in this test is connected to a fix resistive load. Hence, a fix DC current is expected. However, output measurement as shown in Figure 4.7 and Figure 4.8 tells a different story. This is likely due to switching of the system itself.

In term of accuracy of the measurement, we it can be verified by the standard deviation that is close to 0. This tells the consistency of the device but not the accuracy. However, this can be done through a series of calibration in the programming part in the future.

4.4 Real Time Monitoring Using Fixed Load

4.4.1 Boost Converter Efficiency

To begin, we start by testing the efficiency on the fix load. The objective of this is to identify the optimum efficiency level in a time varying scale.

In the appendices shows a series of graph for that shows the measurement of input current, input voltage, output current, output voltage and efficiency using supercapacitor that is charged to 5.0V and load from 25 Ω, 50 Ω, 100 Ω, 220 Ω and 440 Ω.

The test is summarize is shown in the table below.

Table 4.4: Summarize of Real Time Simulation

Load, Ω	Duration of effective discharging, s	Duration of monitoring, s	Average Operating Efficiency, %
25	0	50.8	25
50	12	254.4	40
100	34.2	786.6	45
220	328.8	660	60
440	876.8	1315.2	70

The system is run till the boost converter is unable to maintain the output voltage level. This is where the boost converter consider fail to operate region. Another area to be look at it is the input current. When the boost converter has reaches its maximum PWM ratio, boosting of voltage is at maximum. Thus, the input current also begins to fall as the converter itself is unable to sustain the boosting. The duration of effective discharging is easily determined by the peak of the "input current" graph in the appendix.

Through the graphs in appendices for load using 100 Ω, 220 Ω and 440 Ω., we notice that this system do not shut down itself when it is unable to maintain its output voltage. Instead, the output voltage begins to drop proportionally with the input voltage. This is indicated from the above table as in duration of effective discharging. When the boost converter has reaches its maximum of boost ratio, the PWM is maintained at its maximum duty ratio at 95%. And this is maintained for the rest of the monitoring duration as the boost converter is unable to operate at desire output level.

On the other hand, for lower resistive load such as 25Ω and 50Ω, the boost converter is unable to boost a high load current for this load right from the beginning.

Thus, the high amount of input current is drawn resulting in immediate drop of capacitor voltage level from 5V to 2.8V and 3.3V respectively.

Finally, the load of the resistor plays an important role in determine the boosting capability. For a high load (440 Ω), the boosting ability is able to boost until the input voltage drop below 2.5V. In contrast, minimum input voltage to boost for 220 Ω load is 3V. As for efficiency, it is concluded that higher load resistance has a higher efficiency. This corresponds to the finding obtain by my partner through simulation.

4.4.2 Boost Converter Auto-Shutdown

Below shows a figure of what we presumed as auto-shutdown of the boost converter when using 220 Ω loads.

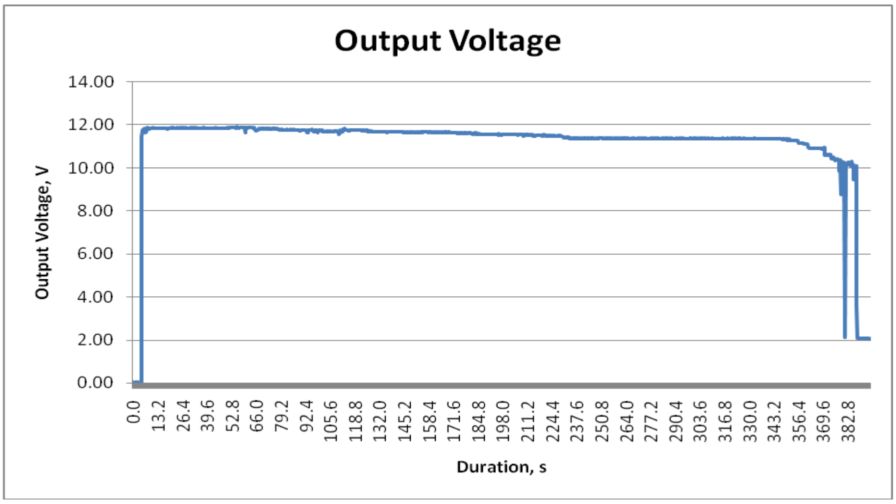


Figure 4.9: Discharging Duration with Timer Malfunction

However, this switch off and unstable period (369.6s to 382.8s) happens when part of the boost converter module misbehave. That module is the timer module.

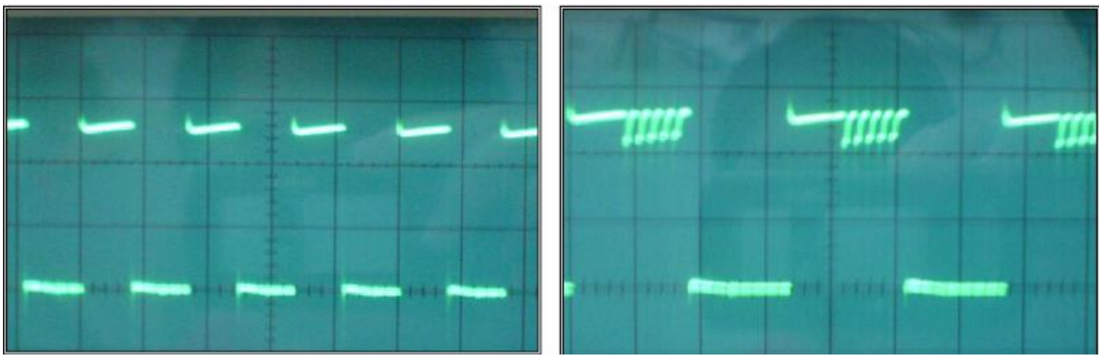


Figure 4.10: Timer Frequency Doubler
(a) Timer Before Breakdown (b) Timer Breakdown (frequency doubler)

This frequency doubler happens then the modulation voltage goes beyond the safe region of 4V. Thus the ripple for Figure 4.10 (b) causes the malfunction of the boost converter as previously indicated as auto-shutdown. Through experiment, this ripple is easily eliminated by providing a higher Vcc voltage (minimum of 6V) to the timer. And the frequency doubler is eliminated.

As for the auto-shutdown mechanism, it is a safer approach to implement comparator as it provides us with a more stable yet predictable shutdown of the system.

4.5 Real Time Monitoring For Motor

4.5.1 Motor Efficiency Monitoring

In this simulation test, the motor is running at freewheeling condition (no load).

The real time simulated motor efficiency is also shown in the appendix. Due to the ability of integrator unable to shut off the boosting when input voltage is below the threshold of 2.3V.

This characteristic is also shown as in the graph for “input voltage using motor running at freewheel” at time 131.4s. A zoom up version is shown as below figure.

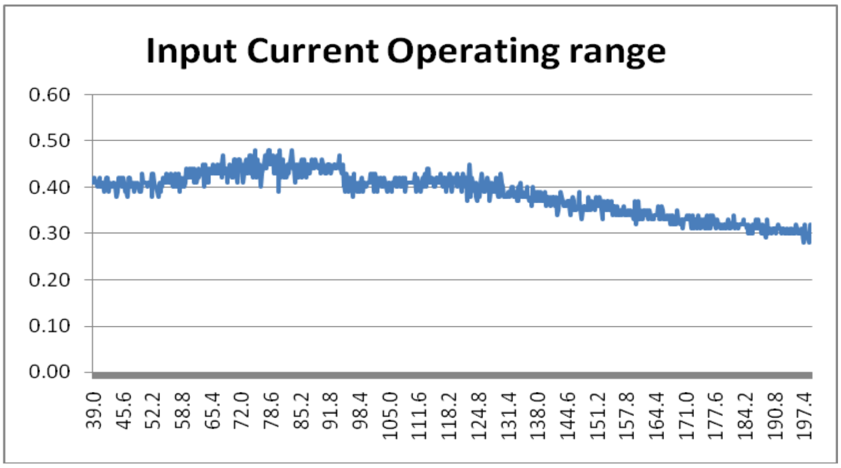


Figure 4.11: Motor Input Current Operating Range

We notice that at 131.4s, the input current is unable to sustain the require input current to run the motor at freewheel. Therefore, while maintain the motor to spin, the motor voltage drops with the input current from the capacitor.

Throughout the whole boosting, we also notice that the efficiency is maintained at an average value of 60%.

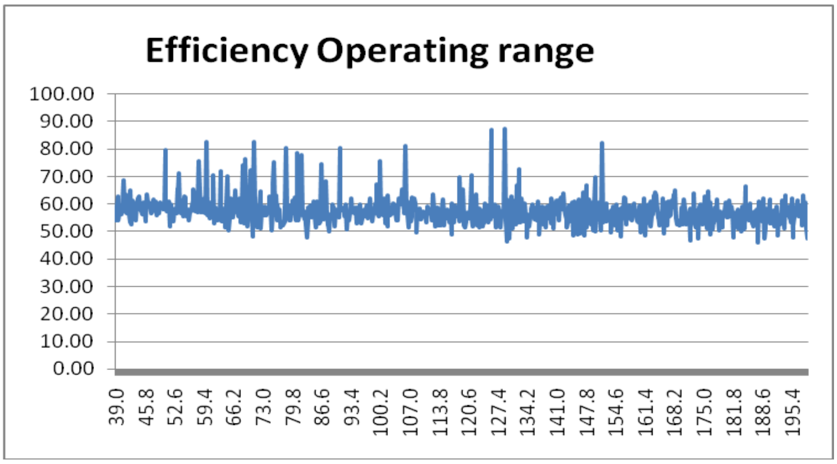


Figure 4.12: Motor Efficiency Operating Range

Though the reading may be slightly inaccurate, the simulated output using this device may still be helpful as it provides us the trend of discharging pattern which allows us for a deeper analysis in the future.

Finally, while conducting test on this system, we notice that at low load, the instant capacitor is connect to the boost converter; the input voltage is drop significantly. As a result, it is noticeable in input voltage graph for 25 Ω, where the starting voltage is at 3.5V instead of 5V. This is due to the high current that is being drawn out from the capacitor.

Therefore, in term of using high current application, we advice that 2 supercapacitor in series at maximum charged of 5.4V is insufficient. In our case, we would require 3 capacitors in series to handle the voltage drop due to current that is being drawn.

4.5.2 Loading Effect of Measurement Device

Below are 2 output voltage graphs obtained using motor as the load, the first figure is the result where Rsense is used, while Figure 4.14 is the result taken without Rsense tapping.

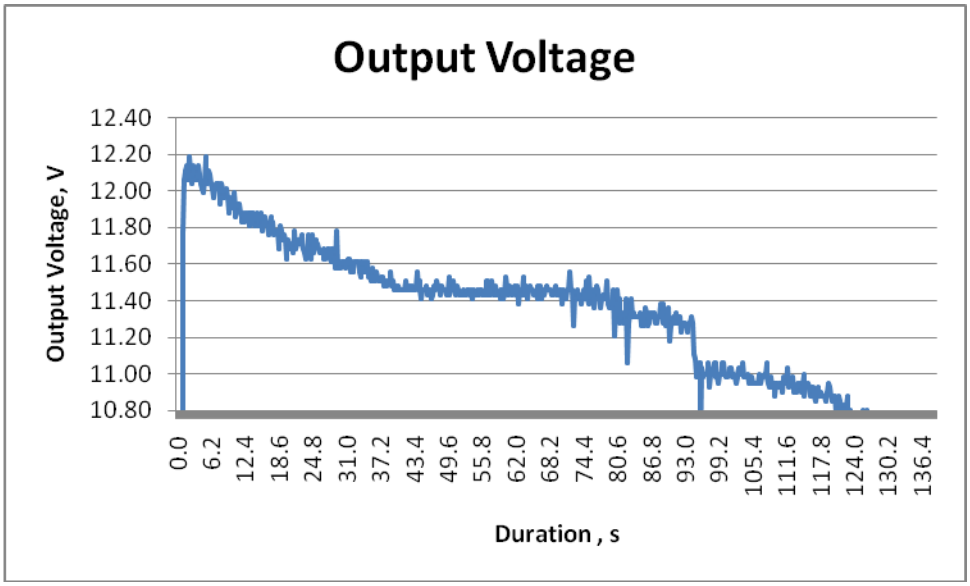


Figure 4.13: Discharge duration with Rsense Loading

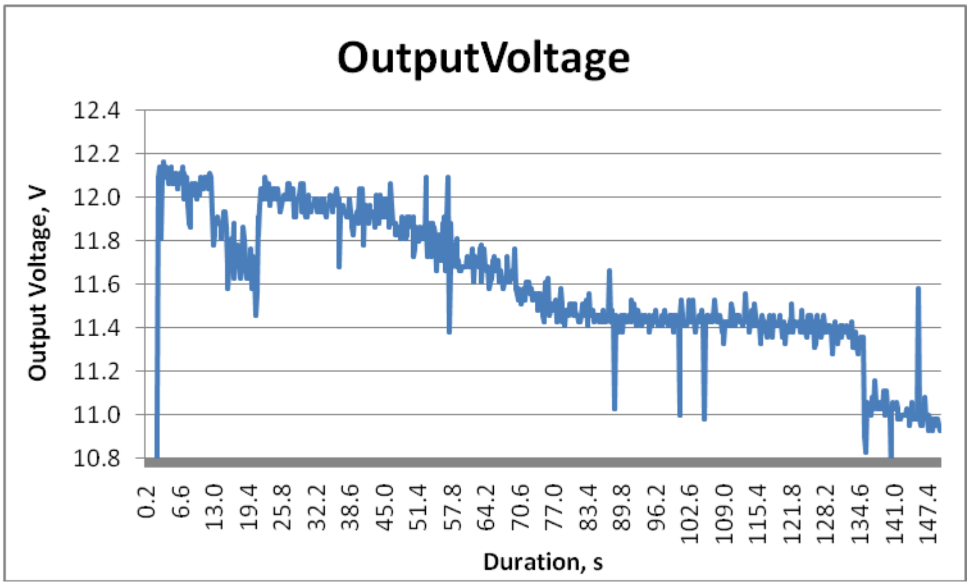


Figure 4.14: Discharge duration without Rsense Loading

The amount of loading effect is illustrated from the 2 figures above. When Rsense is included, the loading effect observed are the decrease in discharging time from 134.6s to 93.0s. This is using output voltage 11.2V as the reference of comparison. Note that the purpose of Rsense is used for current measurement. Thus, efficiency measurement comparison is unable to be done.

The amount of reduction is approximately 31% considering the duration of operation before and after including Rsense. However, in terms of efficiency comparison, it does not affect much toward the time varying value. However, a noticable average value is observe comparing to the experimented value obtained from my project partner.

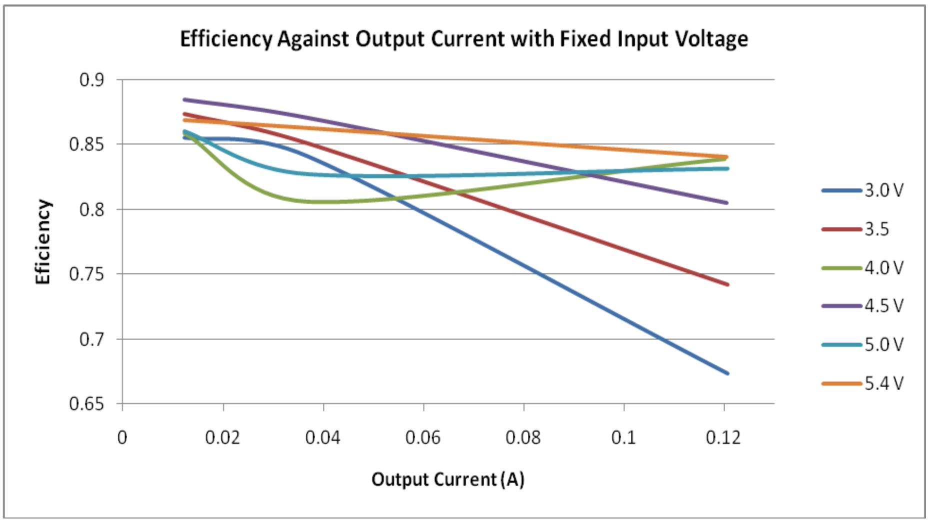


Figure 4.15: Tested Boost Converter Efficiency with Varying Current

Note that the overall efficiency of the circuit is above 80% at high load resistance. However, that is a noticeable decrease of overall efficiency to approximately 65% with the added of Rsense loading effect. This sums it to a massive 15% in reduction of overall efficiency. Thus, this is a major area for further improvement to reduce the loading effect.

4.6 Capacitor Performance

With all the data generated and gathered, the final evaluation will be evaluating the performance of the capacitor through analysis. With the data gather, the tabulated performance of the capacitor is shown as below.

Table 4.5: Energy Conversion for Capacitor Performance

Load	Vstart	Vend	Regulation Time, s	energy drawn, J	Efficiency
100	5	4.3	34.2	162.75	0.302599
220	5	3	328.8	400	0.538036
440	5	2	876.8	525	0.546577
Motor	5	3.5	127.2	318.75	0.478871

The efficiency above is derived from the equation below:

$$E_{AVE} = \frac{P_O}{P_I} = \left[V^2_{O(Average)} * \frac{T_{REGULATION}}{Load\ Resistance}\right] / [{}^1/_2\, CV^2_{INITIAL} - {}^1/_2\, CV^2_{FINAL}]$$

This is be determine the total energy consumed by knowing the duration of regulation, and then divides with the total energy reduced from the capacitor.

For the motor, the load resistance is estimated by using ohm law where R = V/I. Since the operating voltage is 12V and current measured is 0.1A, resistance is obtained from the variables.

In this section, we are focusing on the total energy is transferred through the boost converter. Once again, it is proven that this practical circuit does have a many losses. The energy transferred from the capacitor is unable to obtain a high efficient converter. As shown in the above table, the efficiency is approximately 50%. This clearly shows the performance of the capacitor discharging is far lower than the expected value. In practical circuit, the simulated result of the boost converter shown in Figure 4.15 (done by project partner) seems to be farfetched. And this is very likely due to the limitation of the circuitry such as switching frequency and inductance ESR.

4.6.1 Supercapacitor Energy Density

Continue from table 4.5, this section will focus on the energy density of the super-cap we have. 2.

$$\text{Total energy} = \tfrac{1}{2} CV^2_{INITIAL} = 625 \text{ J (Nominal)}$$

Table 4.6: Supercapacitor Energy Density

Load	Energy drawn, J	Consumption Factor	Energy Density, J/kg	Energy Density, Wh/kg
100	162.75	0.26	3875.000	1.076
220	400	0.64	9523.809	2.645
440	525	0.84	12500.000	3.472
Motor	318.75	0.51	7589.285	2.108

Consumption factor is needed as the total energy drawn from our circuit only range within the operating input range. Therefore, to compensate the uncounted energy, formula below is used.

$$\text{Consumption factor} = \frac{\text{Energy Drawn}}{\text{Total energy}}$$

$$Effective\ Energy\ Density = \frac{Total\ Energy \times Consumption\ Factor}{Capacitor\ Weight}$$

Given that each supercapacitor is 21g (total is 0.042 kg). The energy density is calculated. Comparing it with Figure 2.1, we can see that the growth of the supercapacitor has not grown much since 2005 where the energy density then is only approximately 1 Wh/kg to 2 Wh/kg .

The effective energy density tells us how much energy within the capacitor is considered useful. The value calculated does not include the energy loss through the boost converter. Thus, throughout the past 6 years, we have already observed the growth of energy density. And this has also proved the growing trend of the supercapacitor in this energy storage industry as a force to reckon with.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In a nut shell, the measuring device for efficiency proves to be a critical tool in terms of analysing performance on real time basis. Though there are many imperfections on measuring device that is created, it still manages to provide us with a useful and consistence result for analysis.

Through the result obtained, the efficiency is approximately to be 40% to 70% depending on the load is used (refer to Table 4.4). For a higher resistance load, the system will yield higher output efficiency.

The accuracy of the measuring device is high with error not more than 5%. Due to the interconnection of Rsense for current sensing, overall efficiency is reduced by 31%. This result is shown using motor as basis of monitoring where we observe a increase in operation time from 93s to 134.6s.

Finally, the supercapacitor energy density is calculated to reach maximum value 3.472 Wh/kg when connected to 440 ohm load, and minimum of 1.067 Wh/kg when connected to 100 ohm load (refer to Table 4.6). This shows us a convincing result that shows the growth of capacitor when compare the energy density with the data is gathered at year 2005, that is approximately 1 Wh/kg (Figure 2.5).

5.2 Recommendation

Throughout this project, the major problem basically arises from the current measurement. Not only does the current sensing need to be invisible to the system, it also needs to perform measurement accurately. As the result of the design, the value is fluctuating due to AC characteristic and sensing resistor that is added to the heat loss and reduced efficiency.

However, most of this can be overcome by designing a circuit that is mean to read or convert AC value into a readable yet stable output voltage.

Also, the sensing resistor is best to remove in the future current sensing circuit, as it actually caused some unnecessary loss, and drop the overall efficiency value. Instead, a resistorless approach (LinFinity Microelectronics, 1998) such as figure below maybe implemented to read the value of input current through inductor current. Though this is not a proven design on eliminating the loading effect yet, thus test should be done to determine the performance and improve on the new design.

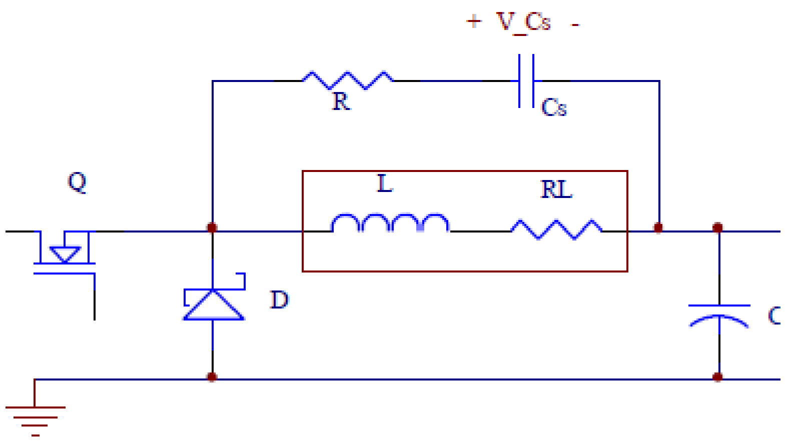


Figure 5.1: Resistorless Current Sensing

Besides that, also notice that the ADC sampling using the average value also not that effective as it also bound to have some numbers of errors. In terms of circuit and hardware modification, I would suggest to use an external oscillator to minimize the jittering effect on the sampling.

To add on, an auto-shutdown of the system could be implemented so that the circuit is immediately shut off when output voltage drop below certain threshold voltage. This can be easily done by adding comparator or silicon control rectifier (SCR) at the output side, since we can predict the fail to operate state for the boost converter.

As for the UART module can be further improve by adding more floating point to allow a more accurate and precise value for analysis.

Finally, a new parameter such as energy density which includes volume should implement as part of the standard for energy storage rating. This is due to the fact that all the value mention before does not disclose any information regarding the size. Through observation, the density (volume/weight) of the capacitor is much lower than battery. And this may not be good for any implementation if the size were to be too big.

5.3 Future Direction

5.3.1 Current Technology

As stated, with the current technology with the support from our result obtained, supercapacitor as already reaches as high as 3.4 Wh/kg. By scaling up the size of this test, how things are headed is rather interesting.

From table 2.5, we note that gasoline have energy density of 12,500 Wh/kg. Assuming 1 litter equals to 1 kg of gasoline, an average town car (Proton Wira) needed 20 kg of petrol to travel 250km. This translate to 2,500 kWh of energy is used. Note that, this uses combustion engine where the efficiency of the energy conversion is low (approximately 13% from chapter 2.4.1, the energy requires 325 kWh to run).

In contrast, taken from an Electric and Hybrid Technology Magazine 2010 issue, it stated that 48kWh of energy is able to translate to about 200-250km of travelling distance for Mercedes-Benz SLS AMG (pure EV for high performance vehicle). Taking our capacitor in comparison for this application, we would require 13,000 kg worth of supercapacitor or 620,000 units of 100F 2.7V of super-capacitor.

Both the example above shows the comparison between fuel and electric driven energy. However, the weight and unit required for electric driven seems rather impractical. Note that, this is solely base of the experimental result we have obtained such as approximately 70% of efficiency and 3.47 Wh/kg.

Therefore, the first area of improvement would be highly base on energy density (Wh/kg) of the super-capacitor. Until then, the consideration of this system to be used of a pure EV is practical to accomplish a reasonable travelling distance.

5.3.2 Future Technology

How this project has led is far beyond comprehensive. Although previous chapter estimation for EV implementation is rather disappointing since the weight issue and quantity require is unattainable. However, as discuss in chapter 4.6.1, we have prove the growing trend of energy density in supercapacitor even at the low end market. This highlighted the huge possibilities of how far capacitor can continue to grow in term of price as well as capacity limit.

Besides that, there are also LIC which has not been factor into, and this has a higher maximum voltage rating of 3.7V. This can also be useful as in increasing the capacity limit. Eventually, it may be compatible or surpassing the current best energy storage, LIB where it is lighter as well as smaller and better in performance.

Also, this may be use for implementation on lighter and smaller vehicle such as 2-seater smart-car. Thus, reducing the requirement and quantity of the energy require onboard.

As for the improvement on the converter, the major room for improvement will be the efficiency of the boost converter. As discuss, due to the low grade component is used, we are unable to reach the desire 100kHz switching frequency, as well as malicious loss (heat and switching loss). Thus, efficiency has been reduced. Furthermore, the Rsense of the current sensing added more loading effect to it, which further lowered the efficiency value of it. Therefore, improvement can be taken on measuring devices as well as boost converter itself, when higher grade component is available or using others topology to reduce loses and loading effect.

REFERENCES

Baisden, A. C., & Emadi, A. (2004). ADVISOR-Based Model of a Battery and an Ultra-Capacitor Energy Source for Hybrid Electric Vehicles. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY* , 199-205.

Battery University. (2005, August). *What is the best battery?* Retrieved Febuary 2011, from http://www.batteryuniversity.com/partone-3.htm

Blick, B. (2005). *Current Sensor*. Retrieved March 26, 2011, from sxlist: http://www.sxlist.com/techref/io/sensor/amps.htm

Chau, K., & Wang, Z. (2005). Overview of power electronic drives for electric vehicles. *HAIT Journal of Science and Engineering B, Volume 2, Issues 5-6* , 737-761.

Hackeray, Thomas, & Whittingham. (2000). Science and Applications of Mixed Conductors for Lithium Batteries. *Materials Research Society* .

Hayt, W., & Kemmerly, J. E. (1971). *Engineering Circuit Analysis (2nd ed.).* McGraw-Hill.

Husain, I. (2003). *Electric and Hybrid Vehicles Design Fundamentals.* Washington, D.C.: CRC Press.

JSR Micro. (2006). *Lithium Ion Capacitor*. Retrieved July 2010, from JSR Micro: http://www.jsrmicro.com/pro_othersIonLithCap.html

Kazimierczuk, M. K. (2008). *Pulse-width Modulated DC-DC Power Converters.* John Wiley & Sons Ltd.

LinFinity Microelectronics. (1998). Simple Current Sense Technique Eliminating a Sense Resistor. *LINFINITY Application Note - AN7* .

Maxim-IC. (2002, July). *App 800: Design a Low-Jitter Clock for High-Speed Data Converters.* Retrieved from Maxim-IC: http://www.maxim-ic.com/app-notes/index.mvp/id/800

Radio Electronics. (2010). Retrieved July 2010, from Capacitor Types and Uses: http://www.radio-electronics.com/info/data/capacitor/capacitor_types.php

Regan, T., Munson, J., Sevastopoulos, A., Zimmer, G., & Stokowski, M. (2005). *App Note 105: Current Sense Circuit Collection.* Linear Technology.

Shah, S. D. (2009). Plug-In Electric Vehicles: What Role for Washington? *(1st ed.). The Brookings Institution.* , pp 29, 37 and 43.

Switching Regulator. (2010). Retrieved July 2010, from National Semiconductor: App Notes: http://www.national.com/analog

Wilson, T. G. (2000). The Evolution of Power Electronics. *IEEE TRANSACTIONS ON POWER ELECTRONICS* , 439-446.

Woodbank Communications Ltd. (2005). *Battery Performance Characteristics*. Retrieved August 2010, from Battery and Energy Technologies: http://www.mpoweruk.com/performance.htm

Zhen, Y. (2010). *AN1332: Current Sensing Circuit Concepts and Fundamentals.* Microchip Technology Inc.

APPENDICES

APPENDIX :

APPENDIX A: Graphs

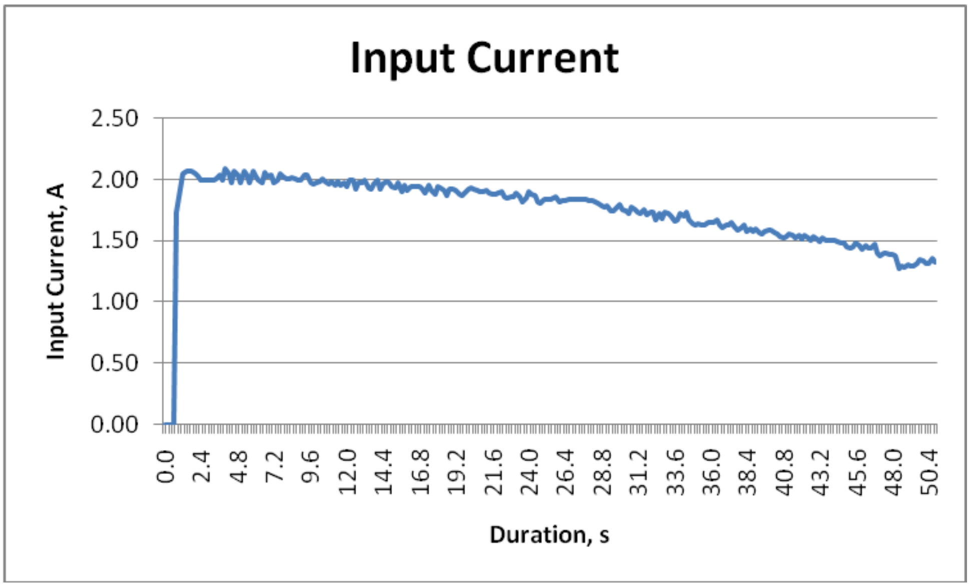


Figure A.1: Input current for load at 25 Ω.

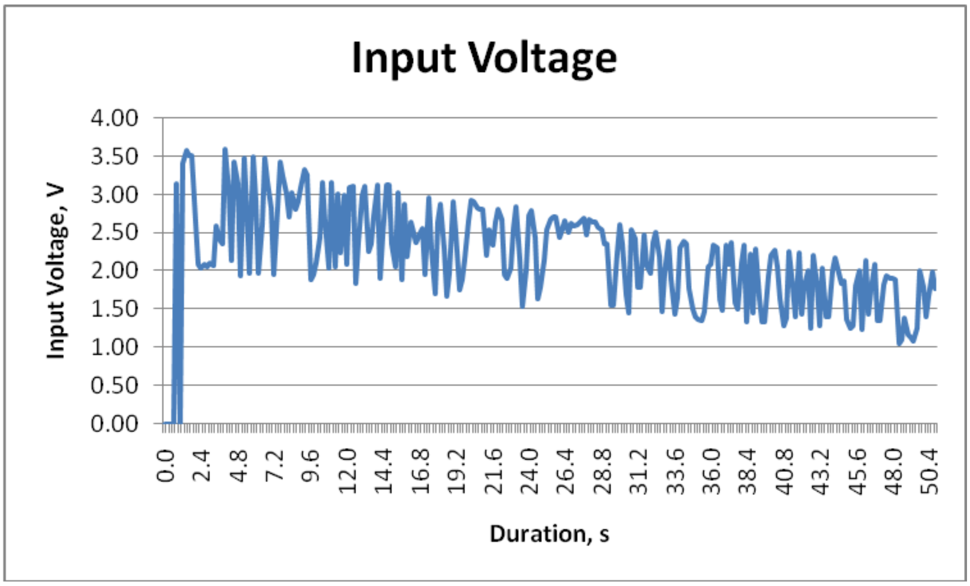


Figure A.2: Input Voltage for load at 25 Ω

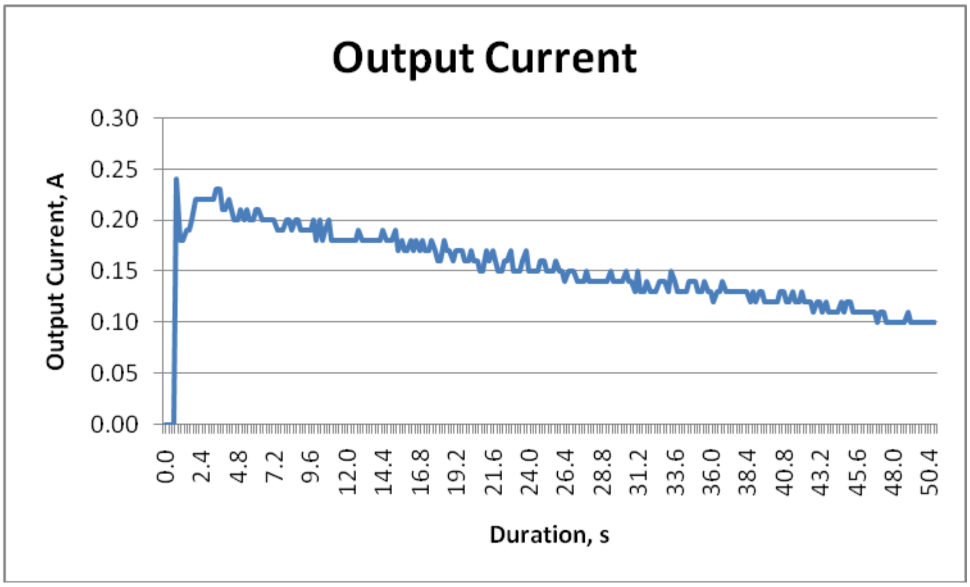


Figure A.3: Output Current for load at 25 Ω

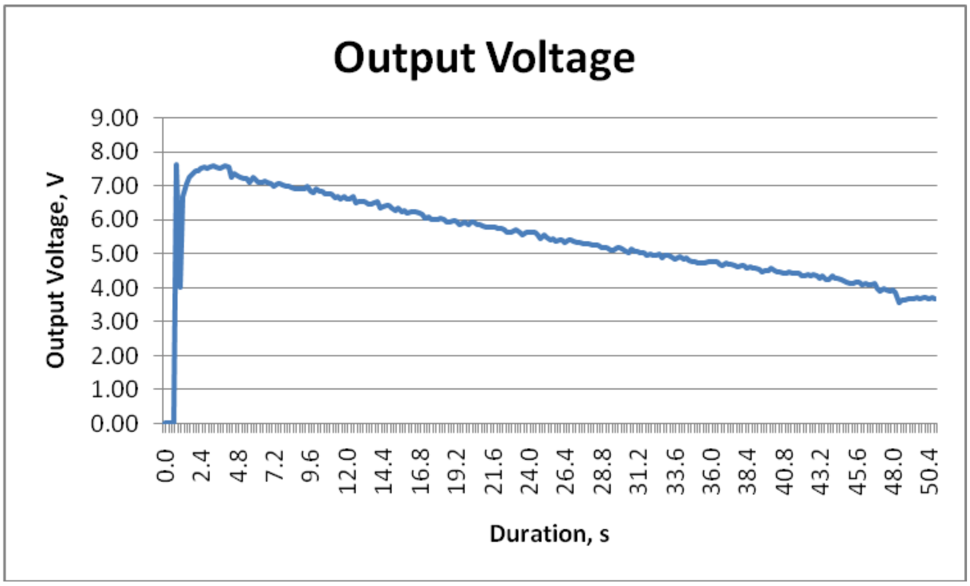


Figure A.4: Output Voltage for load at 25 Ω

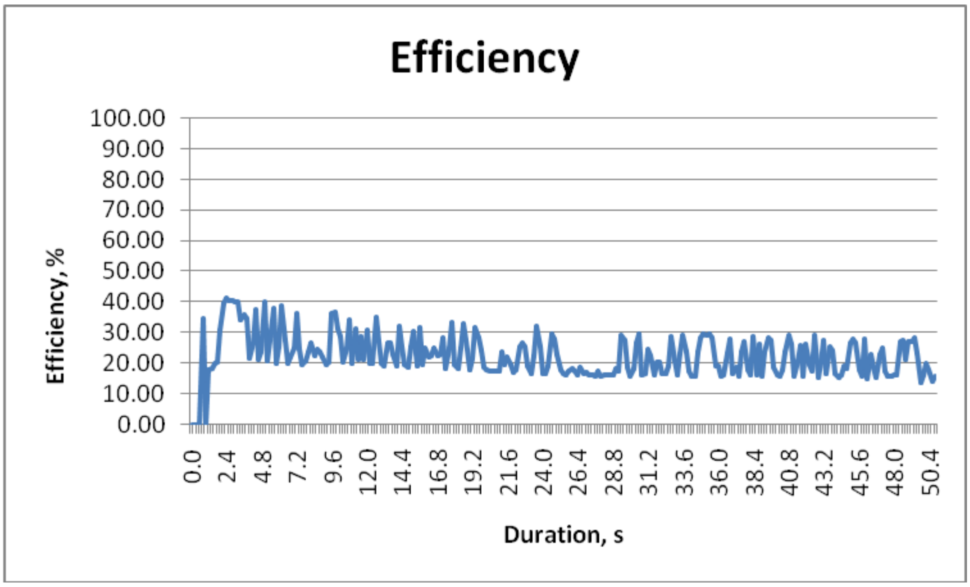


Figure A.5: Efficiency for load at 25 Ω

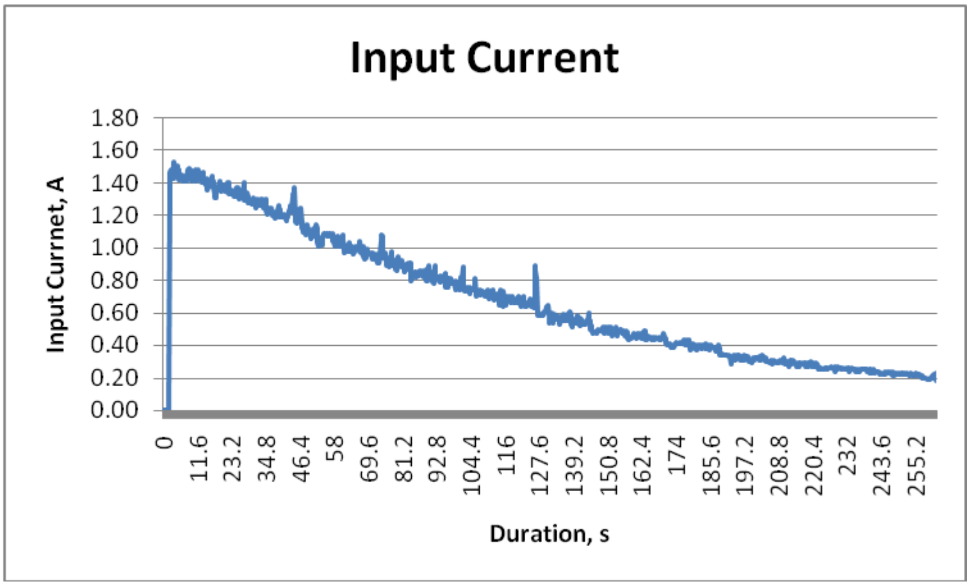


Figure A.6: Input Current for load at 50 Ω

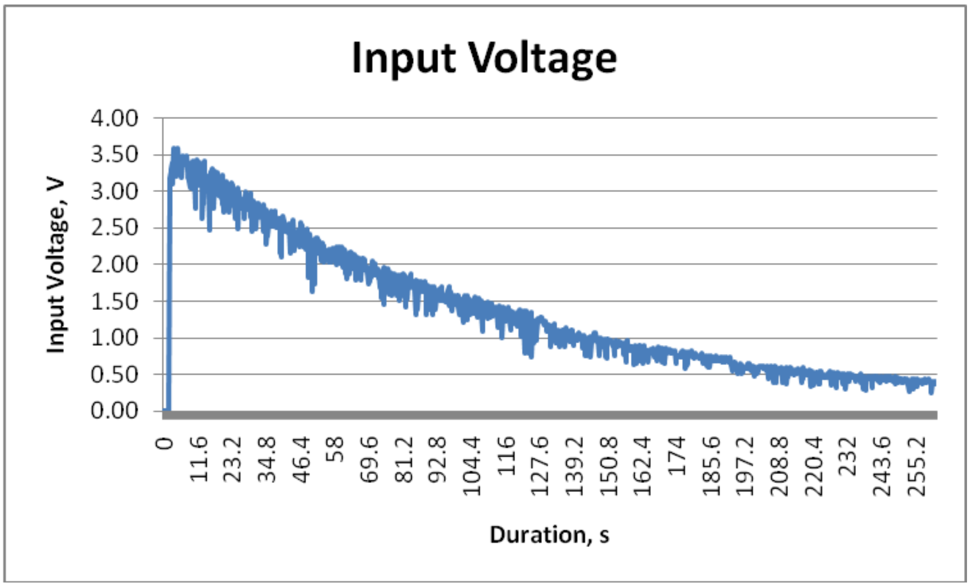


Figure A.7: Input Voltage for load at 50 Ω

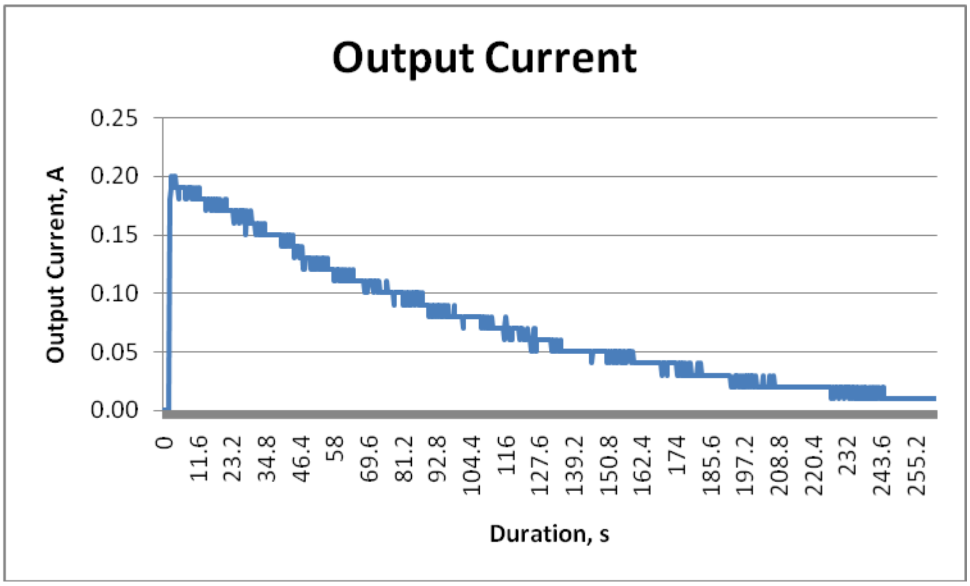


Figure A.8: Output Current for load at 50 Ω

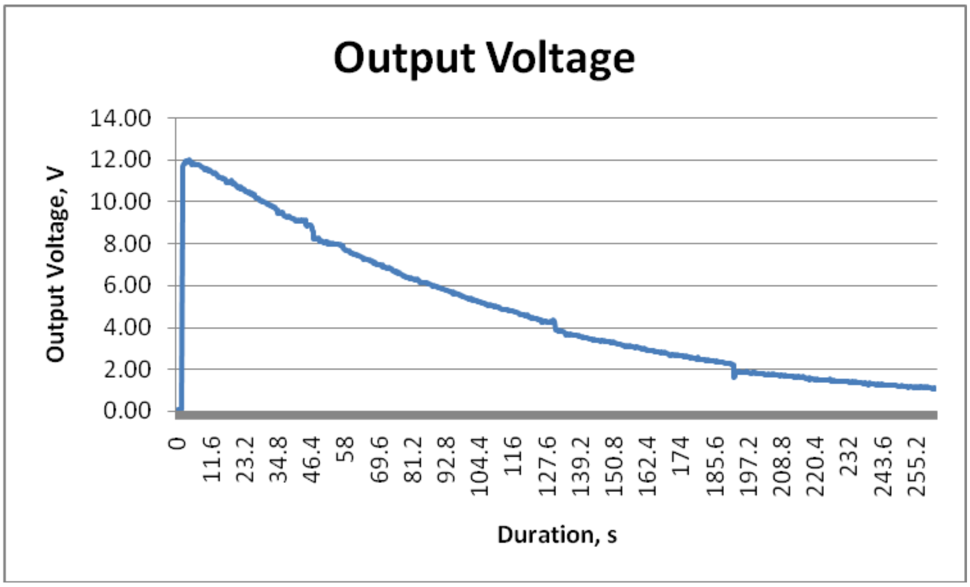


Figure A.9: Output Voltage for load at 50 Ω

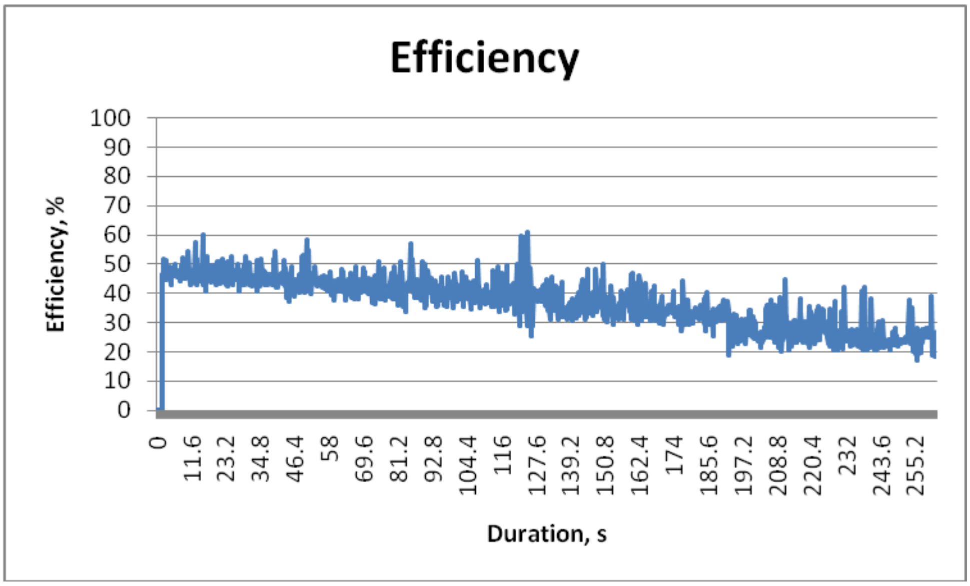


Figure A.10: Efficiency for load at 50 Ω

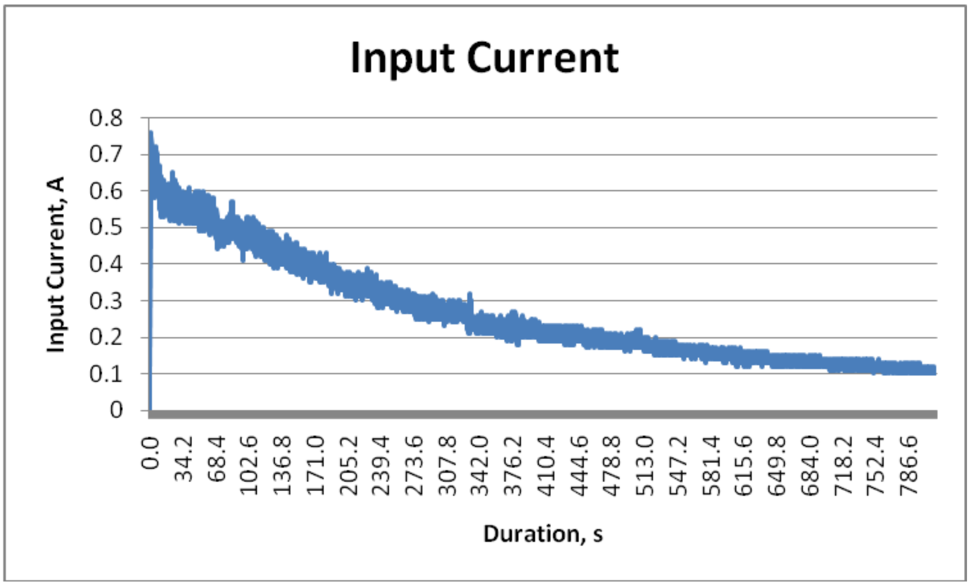


Figure A.11: Input Current for load at 100 Ω

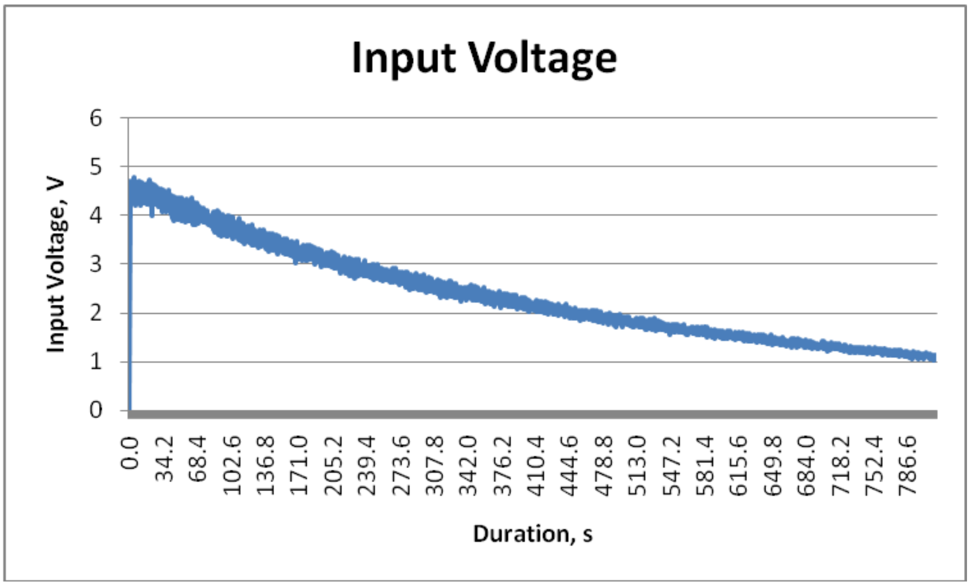


Figure A.12: Input Voltage for load at 100 Ω

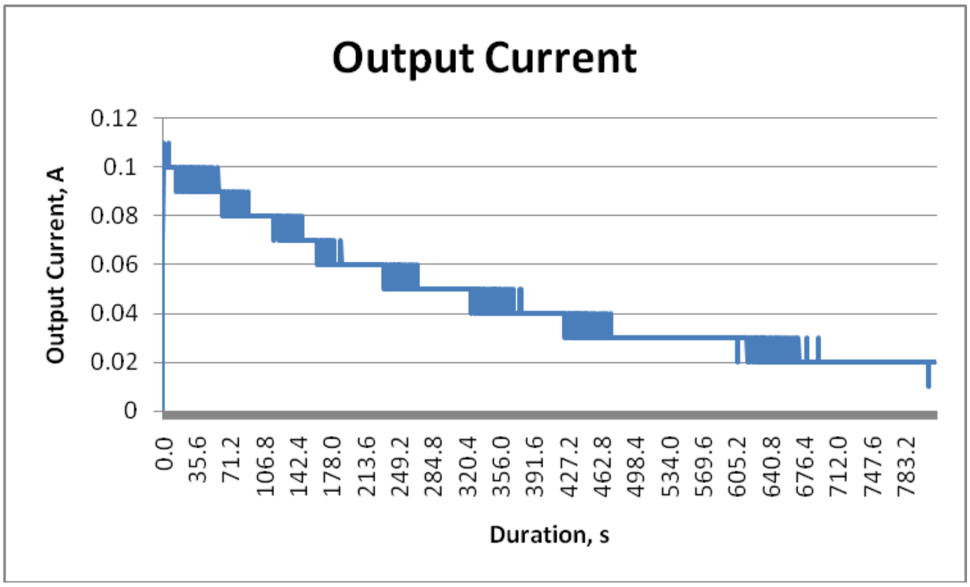


Figure A.13: Output Current for load at 100 Ω

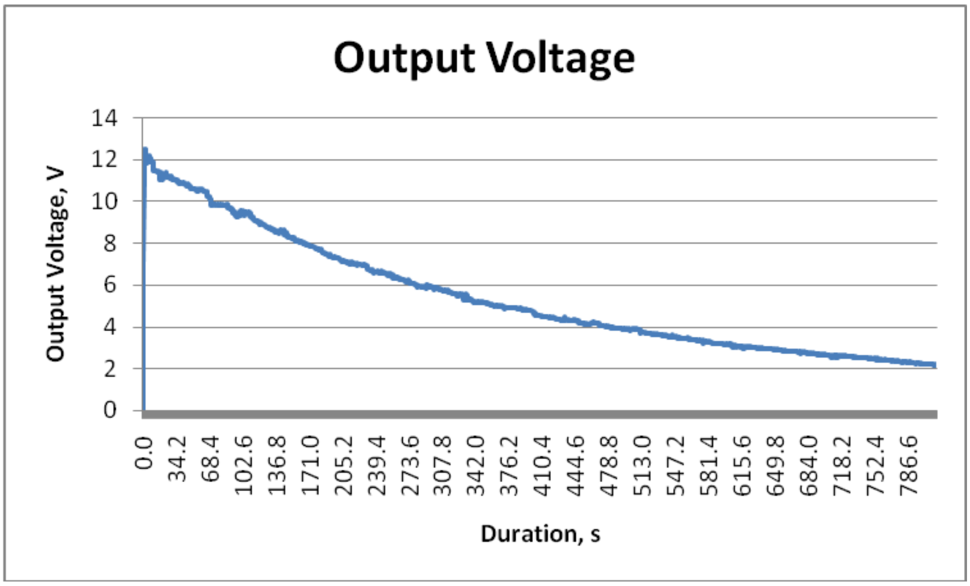


Figure A.14: Output Voltage for load at 100 Ω

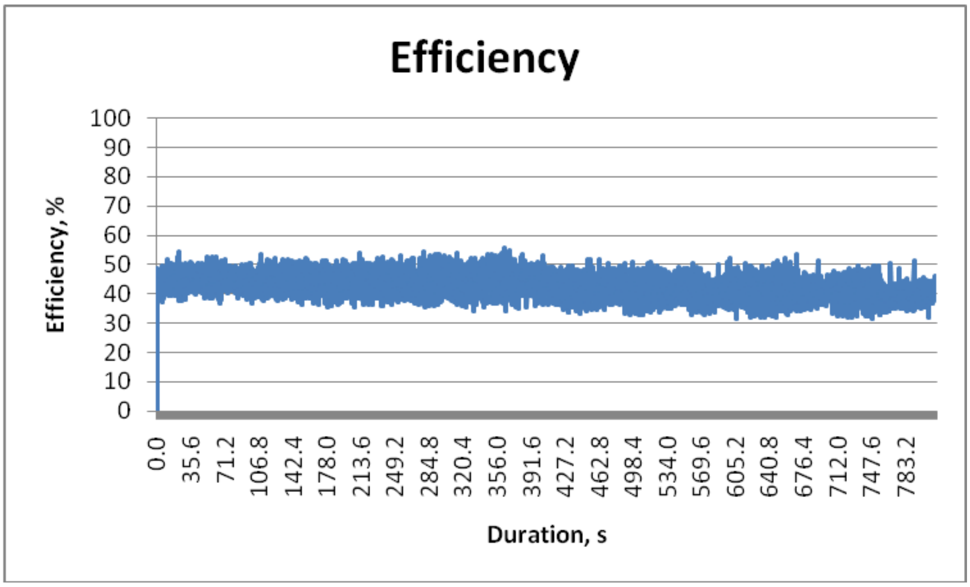


Figure A.15: Efficiency for load at 100 Ω

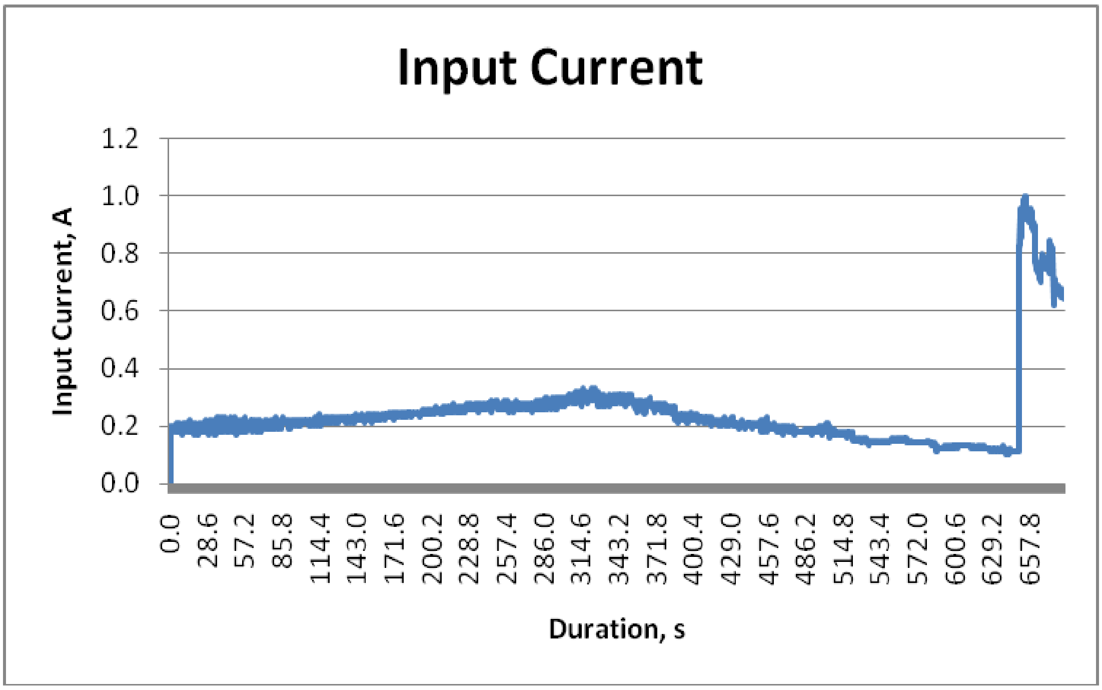


Figure A.16: Input Current for load at 220 Ω

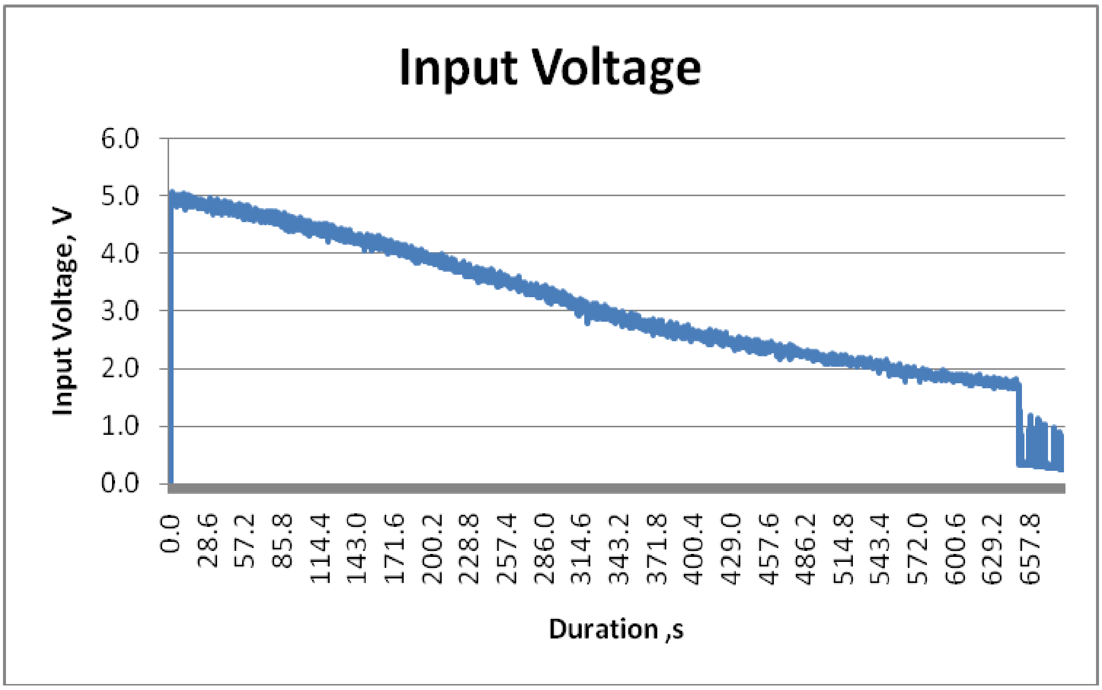


Figure A.17: Input Voltage for load at 220 Ω

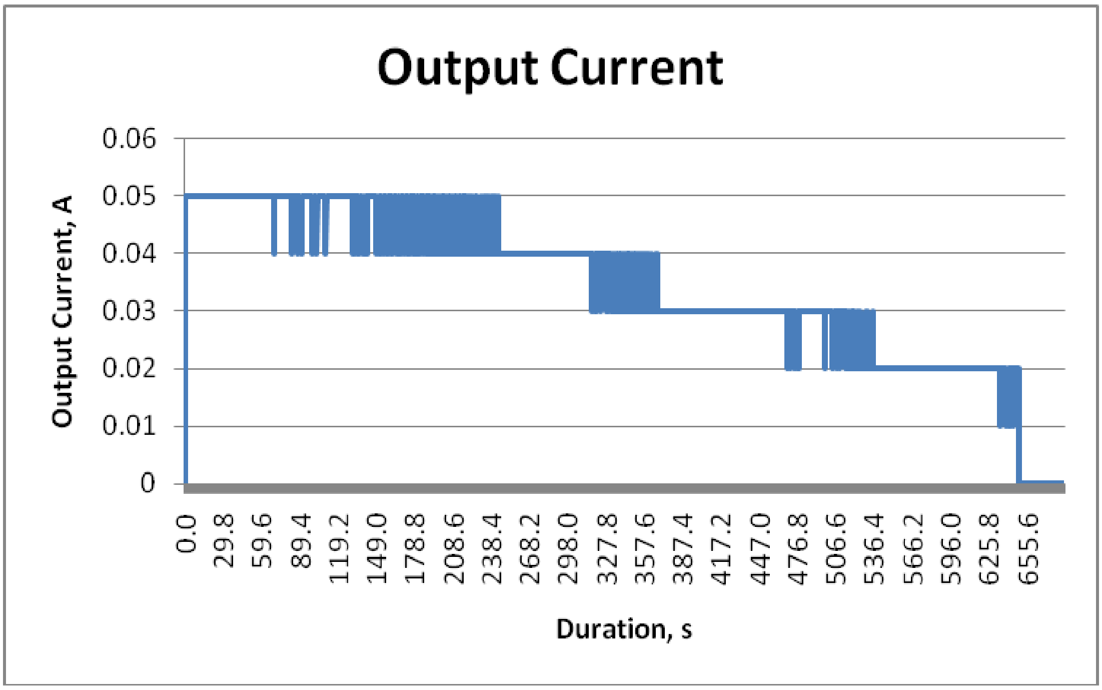


Figure A.18: Output Current for load at 220 Ω

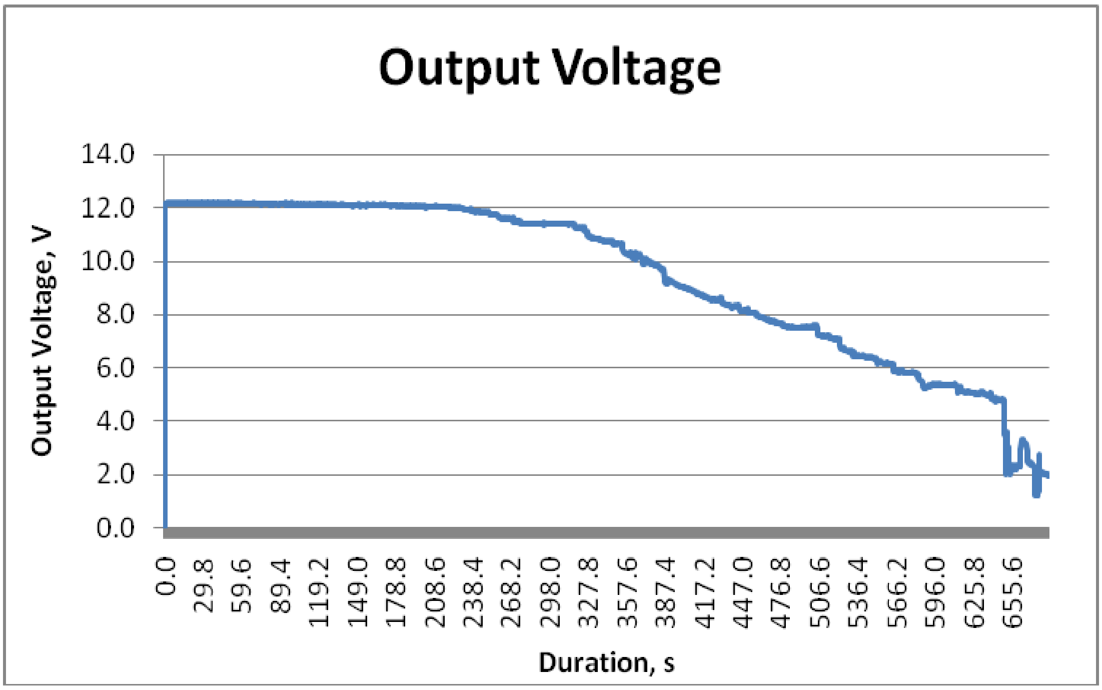


Figure A.19: Output Voltage for load at 220 Ω

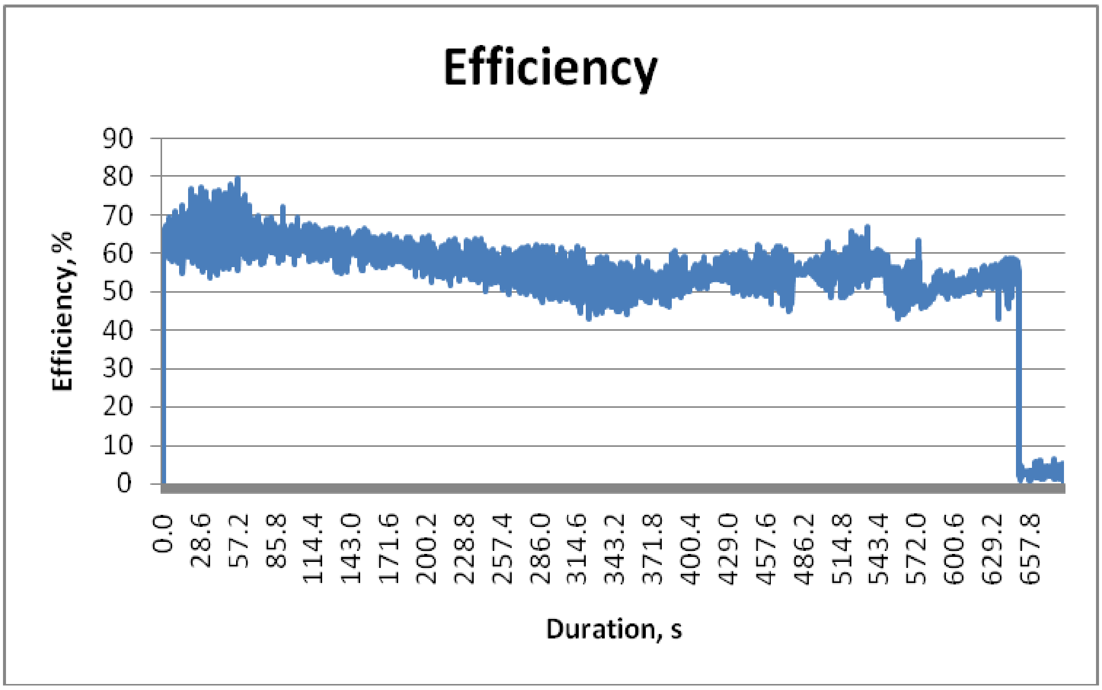


Figure A.20: Efficiency for load at 220 Ω

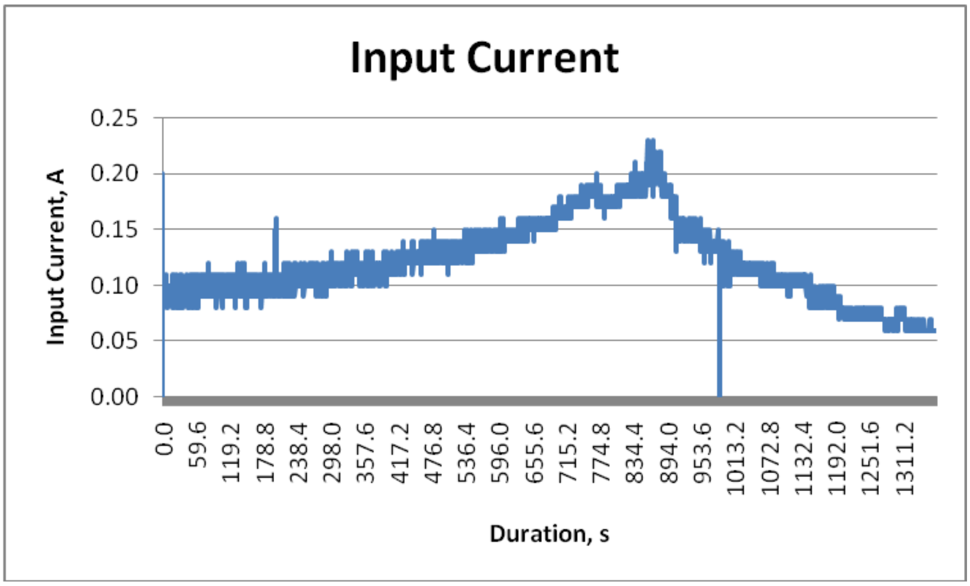


Figure A.21: Input Current for load at 440 Ω

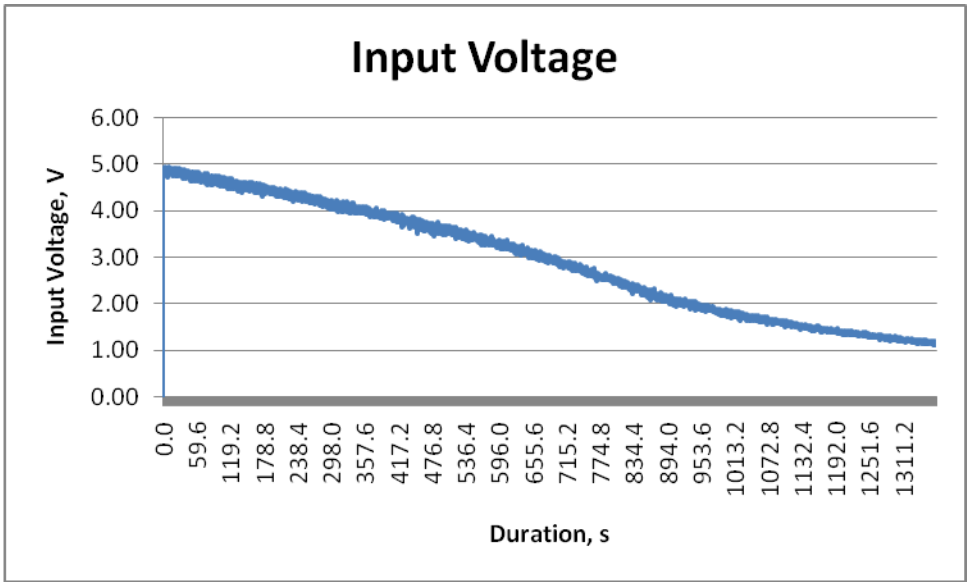


Figure A.22: Input Voltage for load at 440 Ω

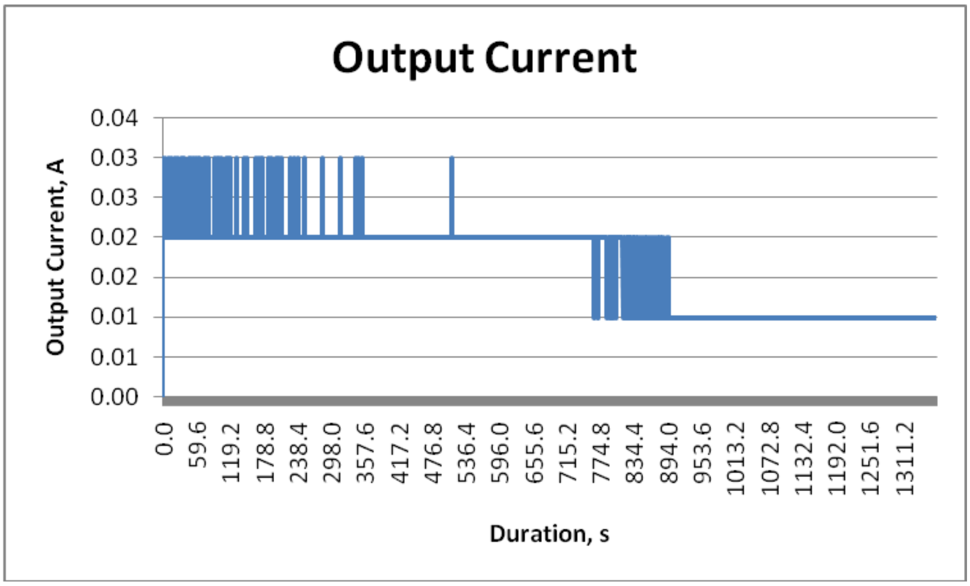


Figure A.23: Output Current for load at 440 Ω

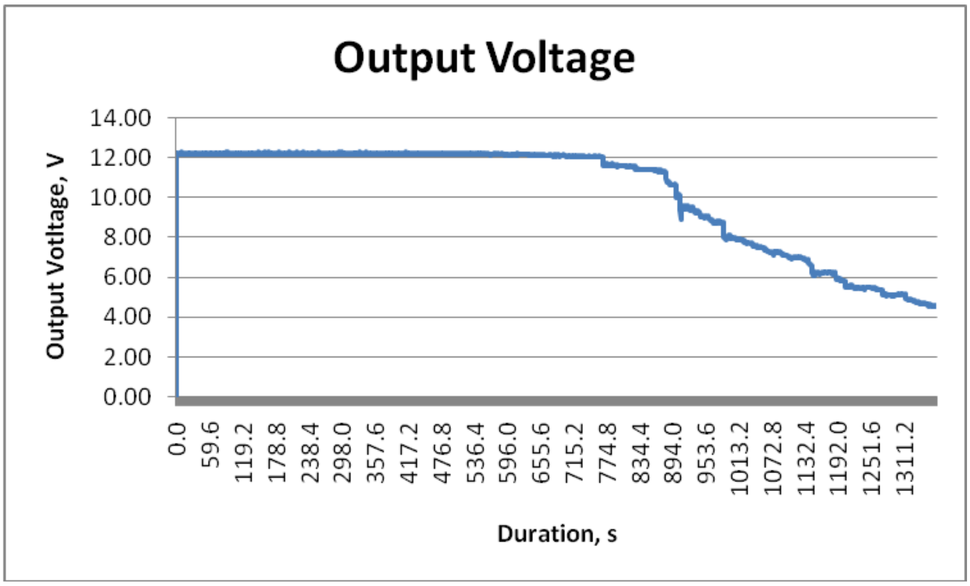


Figure A.24: Output Voltage for load at 440 Ω

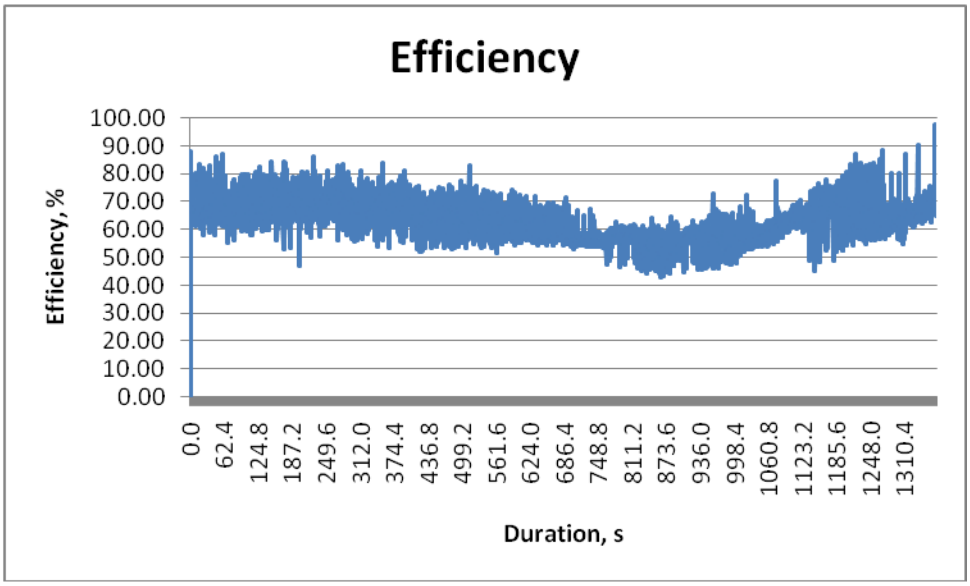


Figure A.25: Efficiency for load at 440 Ω

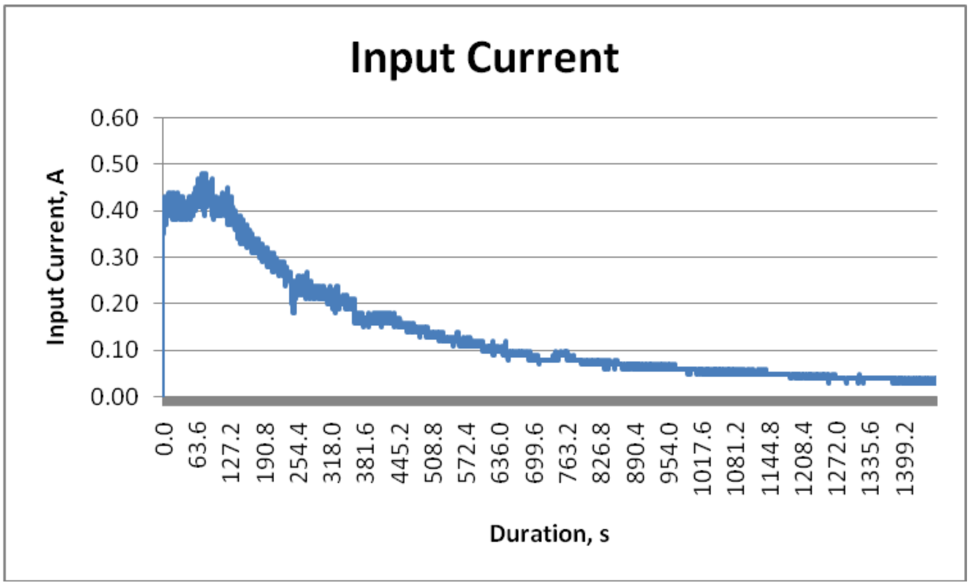


Figure A.26: Input Current for Motor running at freewheel

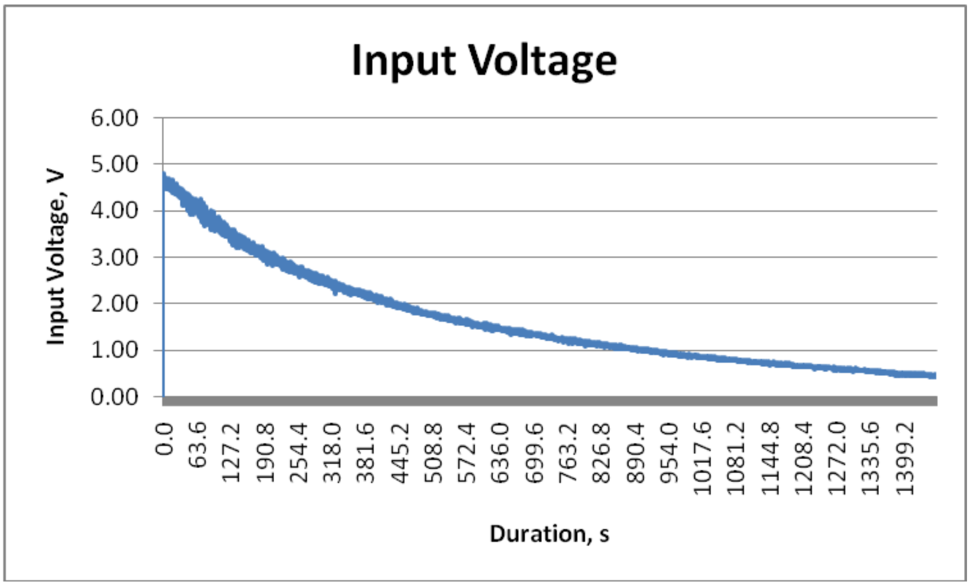


Figure A.27: Input Voltage for Motor running at freewheel

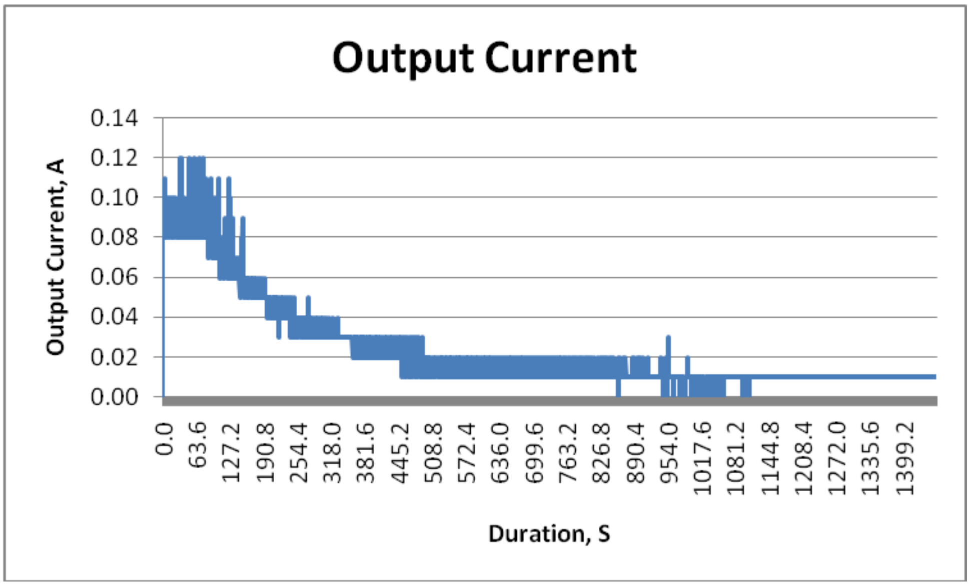


Figure A.28: Output Current for Motor running at freewheel

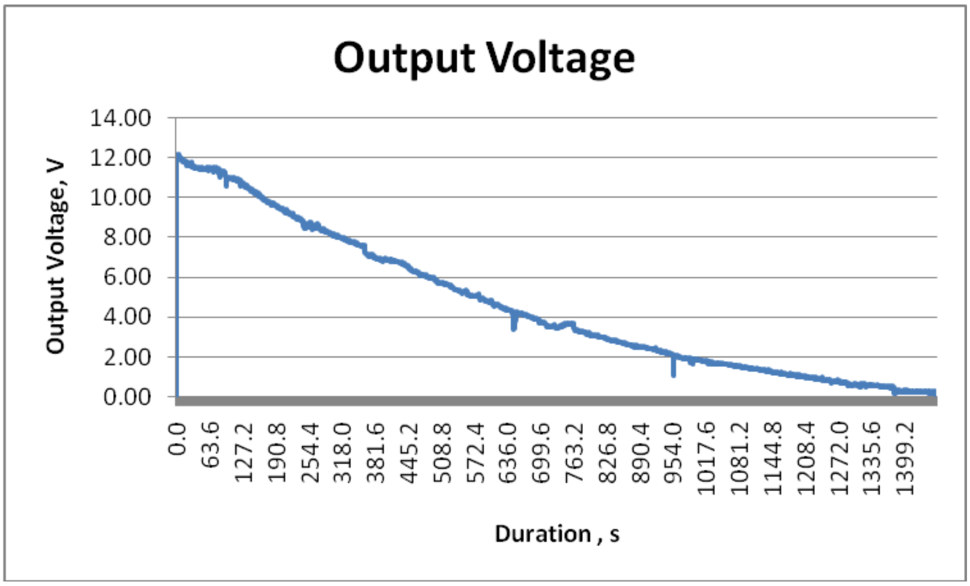


Figure A.29: Output Voltage for Motor running at freewheel

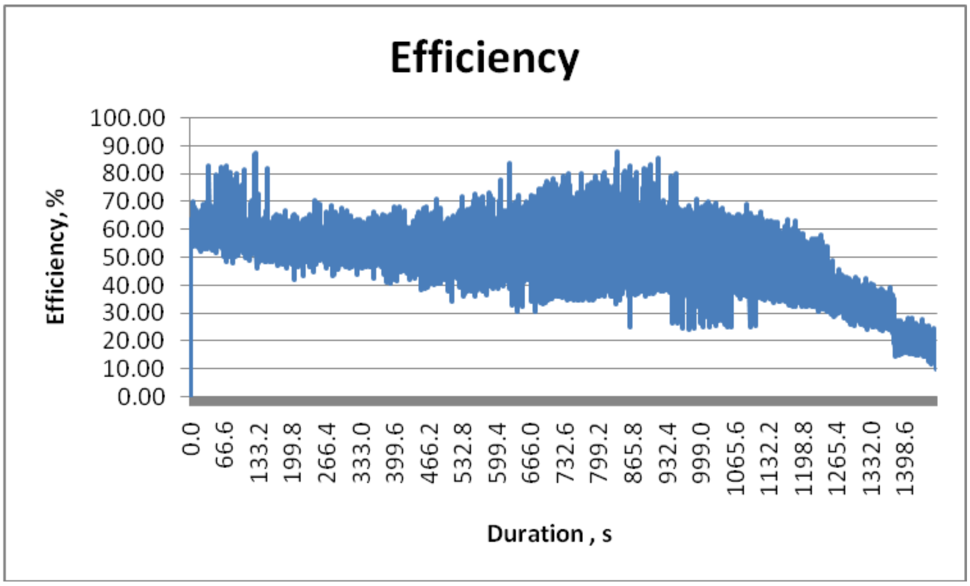


Figure A.30: Efficiency for Motor running at freewheel

APPENDIX B: PCB schematic

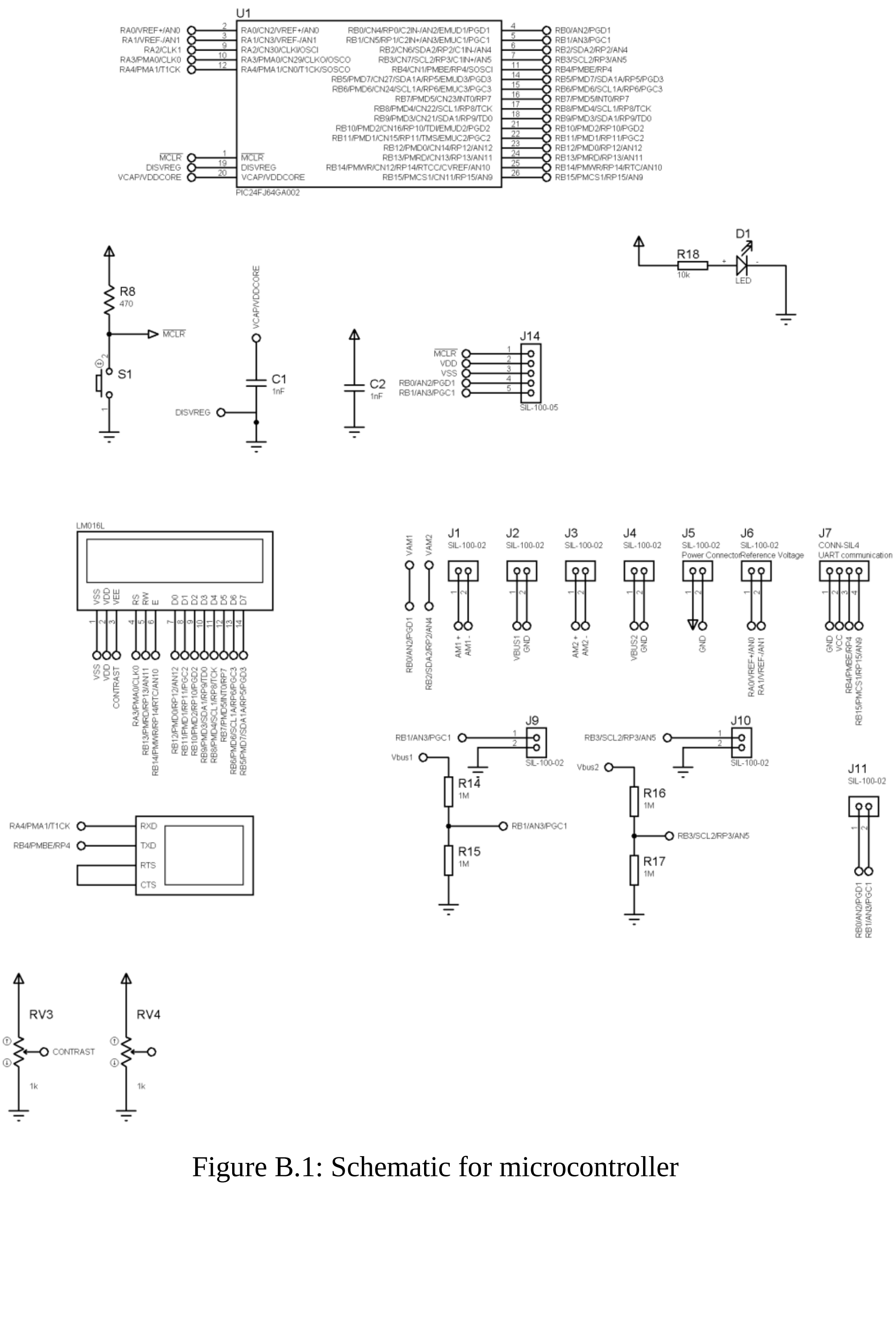


Figure B.1: Schematic for microcontroller

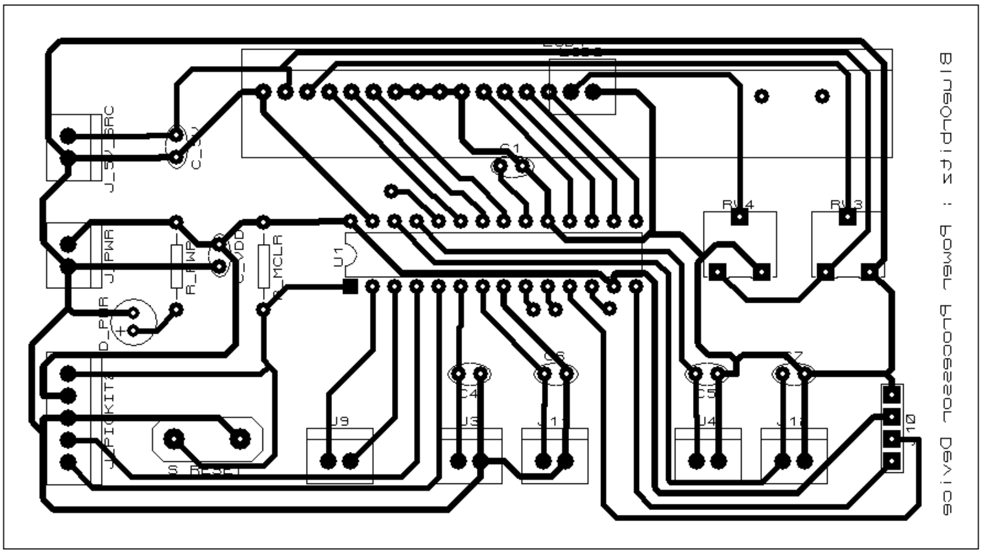


Figure B.2: Microcontroller’s PCB

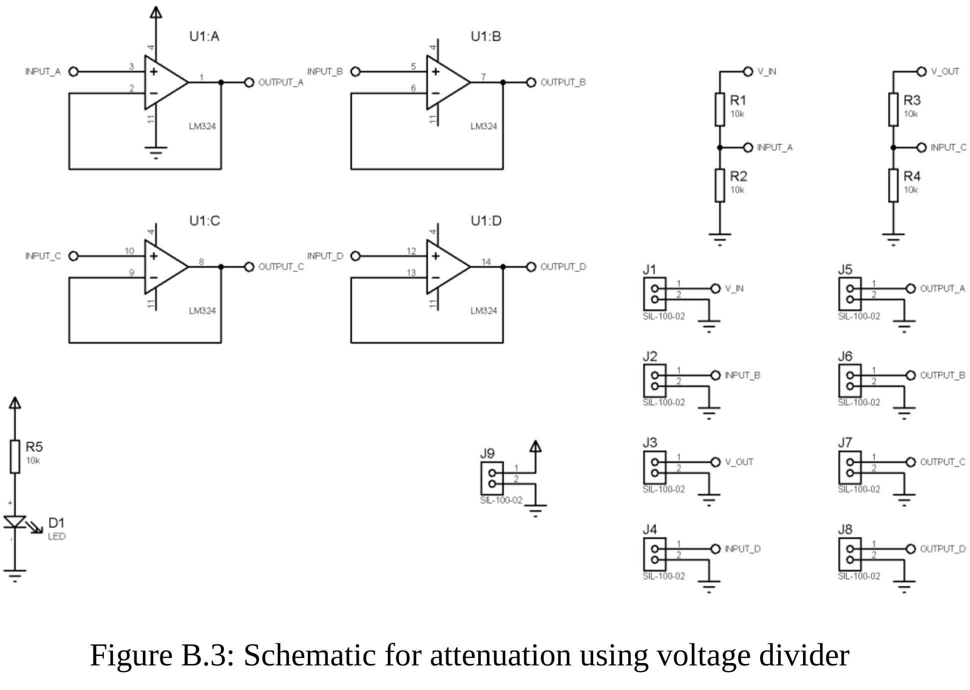


Figure B.3: Schematic for attenuation using voltage divider

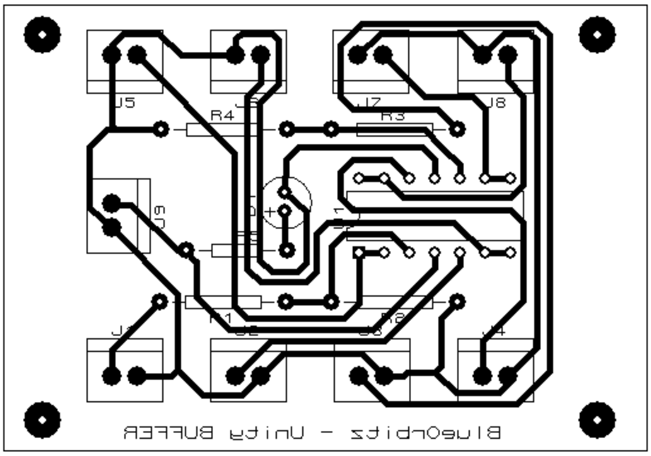


Figure B.4: Attenuation and Buffer PCB

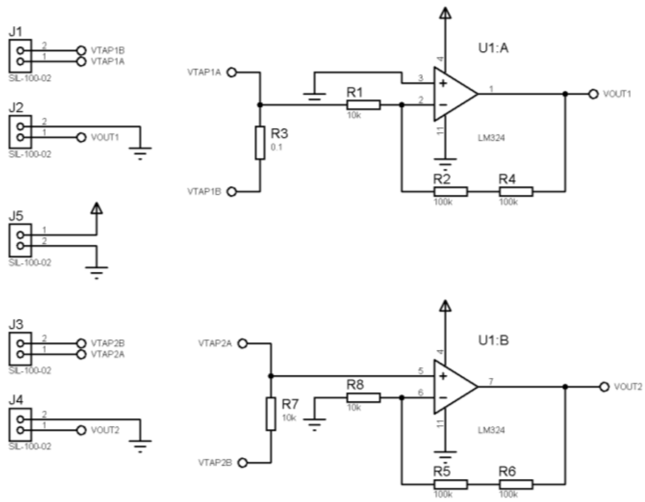


Figure B.5: Current Sensing Schematic

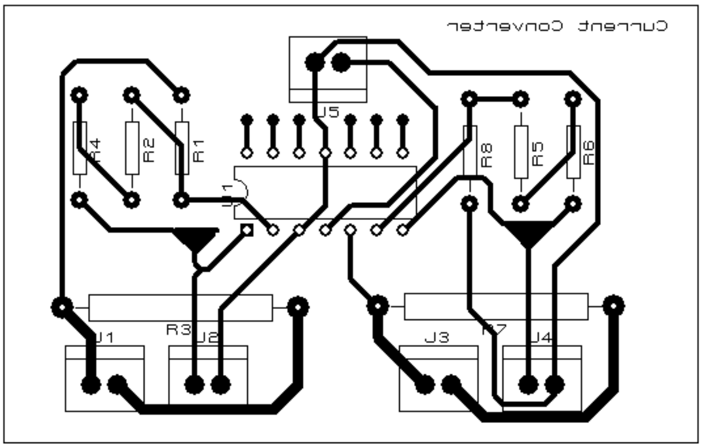


Figure B.6: Current Sensing PCB

APPENDIX C: Program code

Apps.c

```
/*****************************************************************************
*                               uC/OS-II
*                          The Real-Time Kernel
*                 (c) Copyright 2006, Micrium, Weston, FL
*                            All Rights Reserved
*
*
*                         Microchip Application Code
*
* File : APP.C
* By   : Eric Shufo
*
* Rev  : NgMS @ 2 Aug, 2010
*        - Generic PIC24F
*****************************************************************************/
#include <includes.h>

/******************************************************************************
*                                   CONSTANTS
*****************************************************************************/

/******************************************************************************
*                                   VARIABLES
*****************************************************************************/
OS_EVENT *DispSem;

OS_STK ADC_Task_Stk[START_TASK_STK_SIZE];
OS_STK UART2_Task_Stk[START_TASK_STK_SIZE];
OS_STK LCD_Task_Stk[START_TASK_STK_SIZE];

float var_II, var_VI, var_IO, var_VO;
float Pin = 0, Pout = 0;
float eff;

/******************************************************************************
*                                FUNCTION PROTOTYPES
*****************************************************************************/

static  void  ADC_Task(void *p_arg);
static  void  UART2_Task(void *p_arg);
static void  LCD_Task(void *p_arg);

/*****************************************************************************
*                           main()
*
* Description : This is the standard entry point for C code.
* Arguments   : none
****************************************************************************/

CPU_INT16S  main (void)
{
    CPU_INT08U  err;
    OSInit();        /* Initialize "uC/OS-II, The Real-Time Kernel" */
    BSP_Init();
```

```
       DispSem = OSSemCreate(1);
       OSEventNameSet(DispSem, "Semaphore", &err);

    OSTaskCreate(ADC_Task, (void *)0,
        (OS_STK *)&ADC_Task_Stk[0], ADC_TASK_PRIO);
    OSTaskCreate(UART2_Task, (void *)0,
        (OS_STK *)&UART2_Task_Stk[0], UART2_TASK_PRIO);
    OSTaskCreate(LCD_Task, (void *)0,
        (OS_STK *)&LCD_Task_Stk[0], LCD_TASK_PRIO);

#if OS_TASK_NAME_SIZE > 11
    OSTaskNameSet(ADC_TASK_PRIO, (CPU_INT08U *)"ADC measurement", &err);
       OSTaskNameSet(UART2_TASK_PRIO, (CPU_INT08U *)"UART2", &err);
       OSTaskNameSet(LCD_TASK_PRIO, (CPU_INT08U *)"LCD", &err);
 #endif

    OSStart();     /* Start multitasking (i.e. give control to uC/OS-II) */

       return (-1); /* Return an error - This line of code is unreachable */
}

static  void  ADC_Task (void *p_arg)
{
       CPU_INT08U  err;
       (void)p_arg;  /* avoid compiler warning */
       int i;
       float var_IIA;

       ADC_init();

       OSTimeDlyHMSM(0,0,2,0);

       while (1) {
              var_IIA = 0;
              for(i=0;i<10;i++)
              {
                     OSSemPend(DispSem, 0, &err);
                     var_IIA += meter_II();
                     err = OSSemPost(DispSem);
              }

              OSSemPend(DispSem, 0, &err);
              var_II = var_IIA/20;
              var_VI = meter_VI();

              var_IO = meter_IO();
              var_VO = meter_VO();

              err = OSSemPost(DispSem);
              OSTimeDlyHMSM(0,0,0,200);
       }
}

static  void  UART2_Task (void *p_arg)
{
       CPU_INT08U  err;
       (void)p_arg;  /* avoid compiler warning */

       UART2Init();

       OSTimeDlyHMSM(0,0,3,0);
       UART2String("Ready/Start");
```

```
	while (1) {
		OSSemPend(DispSem, 0, &err);
		Pin = var_II * var_VI;
		Pout = var_IO * var_VO;

		if(Pin != 0)
			eff = (float)(Pout/Pin)*100;
		else
			eff = 199.99;

		UART2PutFloat(var_II); UART2PutChar(' ');
		UART2PutFloat(var_VI); UART2PutChar(' ');
		UART2PutFloat(var_IO); UART2PutChar(' ');
		UART2PutFloat(var_VO); UART2PutChar(' ');
		if(eff > 199)
		{
			UART2String("Invalid");
		}
		else
		{
			UART2PutFloat(eff); UART2PutChar('\r');
		}
		err = OSSemPost(DispSem);
		OSTimeDlyHMSM(0,0,0,200);
	}
}

static void LCD_Task (void *p_arg)
{
	CPU_INT08U  err;
	(void)p_arg;

	LCD_init();
	LCD_write(CLEAR_SCREEN, LCD_CMD);
	LCD_center("LCD Display");
	LCD_curpos(LCD_ROW_1, 0);
	LCD_center("Loading...");

	OSTimeDlyHMSM(0,0,3,0);

	LCD_write(CLEAR_SCREEN, LCD_CMD);

	while(1){
		OSSemPend(DispSem, 0, &err);
		LCD_write(CLEAR_SCREEN, LCD_CMD);
		LCD_string("EFF  : ");
		if(eff > 199)
			LCD_string("invalid");
		else
		{
			LCD_sendFloat(eff);
			LCD_sendData('%');
		}

		LCD_curpos(LCD_ROW_1, 0);
		LCD_string("Vout : ");
		LCD_sendFloat(var_VO);
		LCD_sendData('V');

		err = OSSemPost(DispSem);
```

```
            OSTimeDlyHMSM(0,0,0,700);
        }
}
```

bsp_adc.c

```
#include "bsp_adc.h"

#define DELAY 0
#define BUFFER_10BitsSize 1024
#define DELAY_MS 1024
void ADC_delay_ms(int n);

void ADC_delay_ms(int n)
{
        unsigned int count = DELAY_MS;
        while(n--)
        {
                count = DELAY_MS;
                while(--count);
        }
}

void ADC_init(void)
{
        _TRISB2 = 1; //corresponding IO pin to set direction
        _TRISB3 = 1;
        _TRISB15 = 1;
        _TRISB14 = 1;
        ADC_CURRENT_INPUT = 0;
        ADC_VOLT_INPUT = 0;
        ADC_CURRENT_OUTPUT = 0;
        ADC_VOLT_OUTPUT = 0;

        AD1CON1bits.SSRC = 7;//Conversion Trigger Source, 0b000 (Internal
counter ends sampling and starts conversion )
        AD1CON2 = 0;         //Vr +/- = Vdd/Vss, Interrupts at the completion of
conversion for each sample/convert sequence 16-word buffer (ADC1BUFn<15:0>),
MUX A input multiplexer settings
        AD1CON2bits.VCFG = 0b011; //use external Vref +/- pin; AD1CON3 =
0x1F02; max sample time = 31Tad, Tad = 2 x Tcy = 125ns >75ns
        AD1CON3bits.ADRC = 0;//Clock derived from system clock
        AD1CON3bits.SAMC = 0x00001; //1 Tad
        AD1CON3bits.ADCS = 0b00000001; //2 TCY
        AD1CSSL = 1;        // no scanning required
        EnableADC1;
}

void ADC_sel_channel(unsigned int channel)
{
        AD1CHS = channel;
}

int ADC_read(void)
{
    AD1CON1bits.SAMP = 1;  // start sampling, automatic conversion will follow
        ADC_delay_ms(DELAY);
    while (!AD1CON1bits.DONE);   // wait to complete the conversion
    return ADC1BUF0;            // read the conversion result
} // readADC
```

```
/****************************************
  SUB TASK 1
****************************************/
float meter_VI(void)
{
	int i;
	unsigned long readValue=0;

	ADC_sel_channel(ADC_SEL_VI);
	for(i = 0; i<20; i++)
	{
		readValue += ADC_read();
	}
	readValue = readValue/20;

	return((((float)readValue/BUFFER_10BitsSize)*(ADC_POS_VREF-
ADC_NEG_VREF))+ADC_NEG_VREF)*ADC_V1_SCALE;
}

/****************************************
  SUB TASK 2
****************************************/
float meter_II(void)
{
	int i;
	unsigned long readValue=0;

	ADC_sel_channel(ADC_SEL_II);
	for(i = 0; i<50; i++)
	{
		readValue += ADC_read();
	}
	readValue = readValue/50;

	return (((((float)readValue/BUFFER_10BitsSize)*(ADC_POS_VREF-
ADC_NEG_VREF))+ADC_NEG_VREF)*ADC_AM1_SCALE)-ADC_AM1_OFFSET;
}

/****************************************
  SUB TASK 3
****************************************/
float meter_VO(void)
{
	int i;
	unsigned long int readValue=0;

	ADC_sel_channel(ADC_SEL_V0);
	for(i = 0; i<20; i++)
	{
		readValue += ADC_read();
	}
	readValue = readValue/20;

	return (((((float)readValue/BUFFER_10BitsSize)*(ADC_POS_VREF-
ADC_NEG_VREF))+ADC_NEG_VREF)*ADC_V2_SCALE);
}

/****************************************
  SUB TASK 4
****************************************/
float meter_IO(void)
```

```
{
        int i;
        unsigned long readValue=0;

        ADC_sel_channel(ADC_SEL_I0);
        for(i = 0; i<50; i++)
        {
                readValue += ADC_read();
        }
        readValue = readValue/100;

        return (((((float)readValue/BUFFER_10BitsSize)*(ADC_POS_VREF-
ADC_NEG_VREF))+ADC_NEG_VREF)*ADC_AM2_SCALE)-ADC_AM2_OFFSET;
}
```

uart2.c

```
#include  <p24Fxxxx.h>
#include "PPS.h"
#include "uart2.h"

#define SYSCLK          16000000
#define BAUDRATEREG2 SYSCLK/64/BAUDRATE2-1 //12

/*****************************************************************************
 * Overview: Setup UART2 module.
*****************************************************************************/
void UART2Init()
{
    // Set directions of UART IOs
        UART2_TX_TRIS = 0;
        UART2_RX_TRIS = 1;

        iPPSOutput(OUT_PPS, OUT_FN_PPS_U2TX); /* map TX to RP4  */
        iPPSInput(IN_FN_PPS_U2RX, IN_PPS);    /* map RX to RP5  */

        U2BRG = BAUDRATEREG2;
        U2MODE = 0;
        U2STA = 0;
        U2MODEbits.UARTEN = 1;
        U2STAbits.UTXEN = 1;
        // reset RX flag
        IFS1bits.U2RXIF = 0;
}

/*****************************************************************************
  * Overview: Wait for free UART transmission buffer and send a byte.
*****************************************************************************/
void  UART2PutChar(char Ch){
    // wait for empty buffer
    while(U2STAbits.UTXBF == 1);
      U2TXREG = Ch;
}

/*****************************************************************************
 * Overview: Check if there's a new byte in UART reception buffer.
*****************************************************************************/
char UART2IsPressed()
{
    if(IFS1bits.U2RXIF == 1)
```

```
        return 1; //data receive
    return 0; //no data receive
}

/*****************************************************************************
 * Overview: Wait for a byte.
*****************************************************************************/
char UART2GetChar(){
       char Temp;
    while(IFS1bits.U2RXIF == 0);
    Temp = U2RXREG;
    IFS1bits.U2RXIF = 0;
    return Temp;
}

/*****************************************************************************
 * Overview: This function converts decimal data into a string
 * and outputs it into UART.
*****************************************************************************/
void  UART2PutDec(unsigned char Dec){
unsigned char Res;
    Res = Dec;

       if(Res/100)
       {
              UART2PutChar(Res/100+'0');
              if((Res%100)<10)
                     UART2PutChar('0');
       }
       Res = Res - (Res/100)*100;

       if(Res/10)
              UART2PutChar(Res/10+'0');
       Res = Res - (Res/10)*10;

    UART2PutChar(Res+'0');
}

/*****************************************************************************
 * Overview: Print 1 line of string character
*****************************************************************************/
void UART2String(char *var)
{
       while(*var)              //till string ends
       UART2PutChar(*var++);  //send characters one by one
       UART2PutChar('\r');
}

void UART2Word(char *var)
{
       while(*var)              //till string ends
       UART2PutChar(*var++);  //send characters one by one
}

void UART2PutFloat(float fl)
{
       int temp;

       temp = (int)fl;
       UART2PutDec(temp);
       UART2PutChar('.');
```

```
        temp = (int)((fl - (int)fl)*100);
        if (temp < 10 && temp >= 0)
        {
                UART2PutDec(0);
                UART2PutDec(temp);
        }
        else if(temp > 100 || temp < 0)
        {
                UART2PutDec(0);
        }
        else
                UART2PutDec(temp);

}
```

bsp_lcd.c

```
#include "bsp_lcd.h"

#define LCD_SIZE  16
#define DELAY_MS  1024

void delay_ms(int n);

static void LCD_reset(void);
static void LCD_delay(void);

void LCD_init(void)
{
        mPORTBOutputConfig(LCD_PORT_OUTPUT_CONFIG);
        LCD_reset();

        while(!LCD_ready());
        LCD_write(SET_4Bits_2Lines_LCD, LCD_CMD);   // 4-bit, 2 lines, 5x8 font
        delay_ms(2);
        while(!LCD_ready());
        LCD_write(CLEAR_SCREEN, LCD_CMD);   // clear display
        delay_ms(1);
        while(!LCD_ready());
        LCD_write(ENTRY_MODE, LCD_CMD);   // incr address, no display shift
        delay_ms(1);
        while(!LCD_ready());
        LCD_write(DISPLAY_ON_CURSOR_OFF, LCD_CMD);
        delay_ms(1);
}

void delay_ms(int n)
{
        unsigned int count = DELAY_MS;
        while(n--){
                count = DELAY_MS;
                while(--count);
        }
}

void LCD_reset(void)
{

        LCD_RS = 0; LCD_RW = 0;     LCD_EN = 0; Nop();
        delay_ms(20);
```

```
	LCD_EN = 1;
	LCD_delay();
	LCD_D7_LAT = 0; LCD_D6_LAT = 0; LCD_D5_LAT = 1; LCD_D4_LAT = 1;
	LCD_delay();
	LCD_EN = 0;
	delay_ms(100);

	LCD_EN = 1;
	LCD_delay();
	LCD_D7_LAT = 0; LCD_D6_LAT = 0; LCD_D5_LAT = 1; LCD_D4_LAT = 1;
	LCD_delay();
	LCD_EN = 0;
	delay_ms(10);

	LCD_EN = 1;
	LCD_delay();
	LCD_D7_LAT = 0; LCD_D6_LAT = 0; LCD_D5_LAT = 1; LCD_D4_LAT = 1;
	LCD_delay();
	LCD_EN = 0;
	delay_ms(1);

	LCD_EN = 1;
	LCD_delay();
	LCD_D7_LAT = 0; LCD_D6_LAT = 0; LCD_D5_LAT = 1; LCD_D4_LAT = 0;
	LCD_delay();
	LCD_EN = 0;
	delay_ms(1);
}

void LCD_delay(void)
{
	Nop();
	Nop();
	Nop();
}

char LCD_read(char reg)
{
	char data = 0;

	mPORTBOutputConfig(LCD_PORT_OUTPUT_CONFIG);
	mPORTBInputConfig(LCD_PORT_INPUT_CONFIG);

	LCD_RS = reg; LCD_RW = 1; LCD_EN = 0;

	LCD_delay(); LCD_EN = 1; LCD_delay();
	data = LCD_D7_PIN << 7;
	data = data | (LCD_D6_PIN << 6);
	data = data | (LCD_D5_PIN << 5);
	data = data | (LCD_D4_PIN << 4);
	LCD_delay(); LCD_EN = 0;

	LCD_delay(); LCD_EN = 1; LCD_delay();
	data = data | (LCD_D7_PIN << 3);
	data = data | (LCD_D6_PIN << 2);
	data = data | (LCD_D5_PIN << 1);
	data = data | LCD_D4_PIN;
	LCD_delay(); LCD_EN = 0;

	return data;
}
```

```
char LCD_ready(void)
{
	#ifdef SIMULATION
	return 1;
	#else
	return ((LCD_read(LCD_CMD) & 0x80) == 0);
	#endif

}

void LCD_write(unsigned char data, char reg)
{
	while(!LCD_ready());
	mPORTBOutputConfig(LCD_PORT_OUTPUT_CONFIG);

	LCD_RS = reg; LCD_RW = 0; LCD_EN = 0;

	LCD_delay(); LCD_EN = 1; LCD_delay();
	LCD_D4_LAT = ((data & 0x10) == 0x10);
	LCD_D5_LAT = ((data & 0x20) == 0x20);
	LCD_D6_LAT = ((data & 0x40) == 0x40);
	LCD_D7_LAT = ((data & 0x80) == 0x80);
	LCD_delay(); LCD_EN = 0;

	LCD_delay(); LCD_EN = 1; LCD_delay();
	LCD_D4_LAT = ((data & 0x01) == 0x01);
	LCD_D5_LAT = ((data & 0x02) == 0x02);
	LCD_D6_LAT = ((data & 0x04) == 0x04);
	LCD_D7_LAT = ((data & 0x08) == 0x08);
	LCD_delay();  LCD_EN = 0;
}


void LCD_sendData(char c)
{
	LCD_write(c, LCD_DATA);
}

void LCD_string(char *var)
{
	while(*var)		//till string ends
	LCD_sendData(*var++);	//send characters one by one
}

void LCD_center(char *var)
{
	int word, i;

	word = strlen(var);
	if(word<16)
	{
		word = word >> 1; //devide by 2
		for(i=1; i< ((LCD_SIZE >> 1)-word); i++)
			LCD_sendData(' ');
	}
	LCD_string(var);
}

void LCD_curpos(char c, int i) //row, position from left
{
	char LCD_data;
```

```
        LCD_data = (SET_DDRAM_ADD | c ) + i;
        LCD_write(LCD_data, LCD_CMD);
        delay_ms(2);
        while(!LCD_ready());
        LCD_write(ENTRY_MODE, LCD_CMD);
        delay_ms(1);
}

/***********************************************
* send integer to LCD
***********************************************/
void LCD_sendDec(unsigned char Dec)
{
        unsigned char Res;
    Res = Dec;

        if(Res/100)
        {
                LCD_sendData(Res/100+'0');
                if((Res%100)<10)
                        LCD_sendData('0');
        }
        Res = Res - (Res/100)*100;

        if(Res/10)
                LCD_sendData(Res/10+'0');
        Res = Res - (Res/10)*10;

    LCD_sendData(Res+'0');
}

/***********************************************
* send Float to LCD
***********************************************/
void LCD_sendFloat(float fl)
{
        int temp;

        temp = (int)fl;
        LCD_sendDec(temp);
        LCD_sendData('.');

        temp = (int)((fl - (int)fl)*100);

        if (temp < 10)
                LCD_sendDec(0);

        LCD_sendDec(temp);
}
```

bsp_adc.h

```
#include  <p24Fxxxx.h>
#include "adc.h"

#define USE_AND_OR

/************************************
*       External Circuit Parameters
************************************/
#define ADC_POS_VREF  3.3
```

```
#define ADC_NEG_VREF  0
#define ADC_AM1_SCALE 1.8
#define ADC_AM2_SCALE 1.20
#define ADC_V1_SCALE  2     //As attenuator
#define ADC_V2_SCALE  7.8 //(6.06*0.6*1.1) //As attenuator
#define ADC_AM1_OFFSET 0
#define ADC_AM2_OFFSET 0

/*************************************
*      Hardware connected pins
*************************************/
#define ADC_CURRENT_INPUT   AD1PCFGbits.PCFG4
#define ADC_VOLT_INPUT      AD1PCFGbits.PCFG5

#define ADC_CURRENT_OUTPUT  AD1PCFGbits.PCFG10
#define ADC_VOLT_OUTPUT     AD1PCFGbits.PCFG9

#define ADC_SEL_II          ADC_CH0_POS_SAMPLEA_AN4
#define ADC_SEL_VI          ADC_CH0_POS_SAMPLEA_AN5
#define ADC_SEL_IO          ADC_CH0_POS_SAMPLEA_AN10
#define ADC_SEL_VO          ADC_CH0_POS_SAMPLEA_AN9

/*************************************
*      ADC function Library
*************************************/
void ADC_init(void);
float meter_VI(void);
float meter_II(void);
float meter_VO(void);
float meter_IO(void);
```

uart2.h

```
/*****************************************************************************
 * DEFINITIONS

*****************************************************************************/
// Baudrate
#define BAUDRATE2          19200
// UART IOs
#define UART2_TX_TRIS   TRISBbits.TRISB5
#define UART2_RX_TRIS   TRISBbits.TRISB4
#define OUT_PPS                   OUT_PIN_PPS_RP5
#define IN_PPS                    IN_PIN_PPS_RP4

//ASCII special character
#define TAB   0x9

/*****************************************************************************
 * Function
*****************************************************************************/
extern void UART2Init();
extern void  UART2PutChar(char Ch);
extern char UART2IsPressed();
extern char UART2GetChar();
extern void  UART2PutDec(unsigned char Dec);
void UART2String(char *var);
void UART2Word(char *var);
void UART2PutFloat(float fl);
```

bsp_lcd.h

```
#ifndef __BSP_LCD_H__
#define __BSP_LCD_H__

#define USE_AND_OR
/***************************************
               Library
***************************************/
#include "p24fxxxx.h"
#include "ports.h"
#include "string.h"

/**************************************
           Command defination
**************************************/
#define CLEAR_SCREEN                0x01
#define RET_HOME                    0x02
#define SET_4Bits_2Lines_LCD        0x28
#define SET_8Bits_2Lines_LCD        0x38
#define ENTRY_MODE                  0x06
#define DISPLAY_OFF_CURSOR_OFF      0x08   //Without clearing DDRAM content
#define DISPLAY_ON_CURSOR_ON        0x0E
#define DISPLAY_ON_CURSOR_OFF       0x0C
#define CURSOR_BLINKING             0x0F
#define SHIFT_DISPLAY_LEFT          0x18
#define SHIFT_DISPLAY_RIGHT         0x1C
#define MOVE_CURSOR_LEFT            0x10
#define MOVE_CURSOR_RIGHT           0x14
#define SET_DDRAM_ADD               0x80   //OR with the using address
#define LCD_ROW_0                   0x00
#define LCD_ROW_1                   0x40


#define LCD_CMD   0
#define LCD_DATA  1

/*************************************
      Hardware connected pins
*************************************/
// IMPORTANT NOTE: BSET/BCLR in PIC24 is R-M-W
// It is better to write to latch than pin
#define LCD_RS _LATB12
#define LCD_RW _LATB11
#define LCD_EN _LATB10

#define LCD_D4_PIN _RB9
#define LCD_D5_PIN _RB8
#define LCD_D6_PIN _RB7
#define LCD_D7_PIN _RB6

#define LCD_D4_LAT _LATB9
#define LCD_D5_LAT _LATB8
#define LCD_D6_LAT _LATB7
#define LCD_D7_LAT _LATB6

#define LCD_PORT_OUTPUT_CONFIG 0x1FC0 //0b 0001 1111 1100 0000
#define LCD_PORT_INPUT_CONFIG  0x03C0 //0b 0000 0011 1100 0000

/************************************************************************
                       LCD function Library
************************************************************************/
```

```c
void LCD_init(void);
void LCD_write(unsigned char data, char reg);
char LCD_read(char reg);
char LCD_ready(void);

void LCD_sendData(char c);
void LCD_string(char *var);
void LCD_center(char *var);
void LCD_curpos(char c, int i);
void LCD_sendDec(unsigned char Dec);
void LCD_sendFloat(float fl);
#endif
```