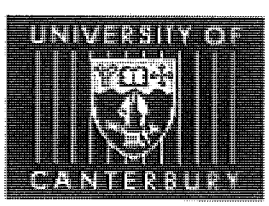
An Auxiliary Power Distribution Network for an Electric Vehicle

A Thesis submitted in partial fulfilment of the requirements
for the degree of Master of Engineering (Electrical and Electronic)
at the
University of Canterbury
Christchurch, New Zealand



By
William Wen-Hou Chen
B.E. (Hon.)

University of Canterbury
Christchurch, New Zealand
31st May 2003

ABSTRACT

The University of Canterbury purchased a modern Toyota MR2 sports car with the intention of converting it into an electric vehicle. Similar to the common combustion engine vehicles, electric vehicles requires power and control systems to operate the 12Vdc auxiliaries, such as lights, indicators and wipers. Traditional technology results in a large number of wires in the wiring harness. To reduce the number of wires, an alternative method is to use a pair of control lines and a universal power connection around the vehicle. This power and control system is named the "Power Distribution Network" and it is implemented by using multiple power converters and a differential control system.

This thesis presents the design, implementation, and test results of the Power Distribution Network for the MR2. The 300Vdc nominal battery voltage is converted to an intermediate voltage of 48V. This configuration is considered more efficient than the usual 12V distribution system since smaller and lighter wires can be used to carry same amount of power. The Power Distribution Network operates off the 48V intermediate voltage, and provides 12V output power to drive all the auxiliaries in the vehicle. The Power Distribution Network also has the ability of detect faults from the auxiliary loads as well as turn on and off these loads.

The Power Distribution Network is implemented with two major systems: the Auxiliary Power System, which consists of a 360W Cuk converter with current limiting control circuits to step-down voltage from the 48V intermediate voltage to the 12V. The other system is the CAN Control system, it is developed using micro-controllers and stand-alone CAN controllers that control and monitor the auxiliary loads in the vehicle.

The prototype Power Distribution Network is fully operational and has been tested with eight of 12V light bulbs which are used to simulate the auxiliary loads in the vehicle. Experimental measurements show that the prototype is able to successfully control the light bulbs under the full load condition. This confirms that in principle the Power Distribution Network is suitable as the power and control system for the auxiliary loads in an electric vehicle.

ACKNOWLEDGMENTS

I would like to express my gratitude to those who have helped me in the course of this thesis. First and foremost, I would like to gratefully thank my supervisors, Dr. Richard Duke and Dr. Simon Round for their guidance, assistance and encouragement throughout the thesis.

Thanks are also extended to the technical staff members: Mr. Ken Smart and Mr. Ron Battersby – for their technical assistance in the laboratory. I would also like to thank Mr. Philipp Hof for his assistance with the software testing and Mr. Nick Smith for his help with the PCBs manufacturing.

I would like to thank the members of power electronics postgraduate group, Dr. Hamish Laird, Tony Rocco, Si Kuok Ting, Irene Ting and Leo Hsieh for providing good advice and a pleasant working environment in the laboratory.

Last but not least, I would like to express my deepest thank to my friend Nicole and my parents David and Mimi for their encouragement and support over the years of study.

PUBLICATION RELATED TO THIS THESIS

William Chen, Richard Duke, Simon Round "Design of an Auxiliary Power Distribution Network for an Electric Vehicle", Delta 2002, the first international workshop conference. Christchurch, New Zealand, Jan 2002.

William Chen, Richard Duke, Simon Round "Design of an Auxiliary Power Distribution Network for an Electric Vehicle", AUPEC 2003, Australian Universities Power Engineering Conference. Christchurch, New Zealand, Sept 2003.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Electric Vehicles have a much longer history than most people realise. Electric vehicles were seen soon after Joseph Henry first introduced the first DC powered motor in 1830. A small model of electric vehicle is built by Professor Stratingh in the Dutch town of Gröningen in 1835. The first full sized electric vehicle was built in 1834 by Thomas Davenport in USA and followed by Moses Farmer who built the first two passengers electric vehicle in 1847. There were no rechargeable electric cells (Batteries) at this time. An electric vehicle did not become a viable option until the Frenchmen Gaston Plante and Camille Faure respectively invented (1865) and improved (1881) the storage battery [1].

Figure 1.1 shows a 1903 Krieger electric car. This car is a front wheel drive electric-gasoline hybrid car and has power steering. A gasoline engine supplements the battery pack. Between 1890 and 1910, there were many hybrid electric cars and four wheel drive electric cars. Electric cars were more expensive than gasoline cars and electrics were considered more reliable and safer [1].

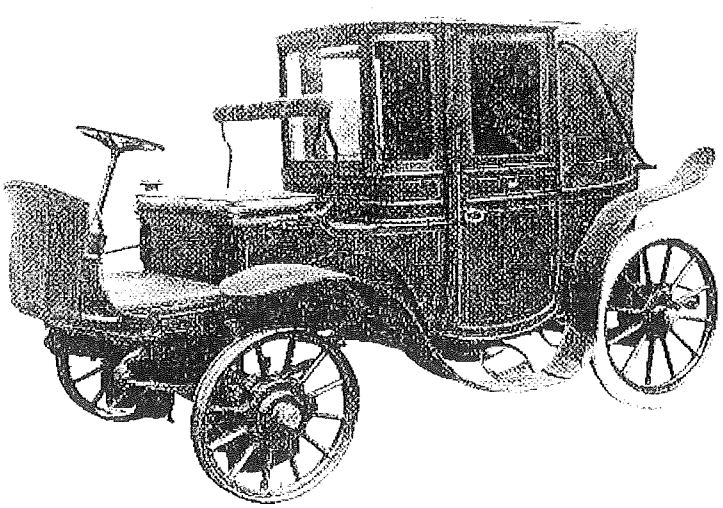


Figure 1.1 1903 Krieger electric car [1].

Electric vehicles are known as Zero Tailpipe Emissions Vehicles, and they produce less pollution than Petrol or LPG powered vehicles. As electric vehicles have fewer moving parts, maintenance is also minimal. With no internal combustion engine there are no oil changes, tune-ups, timing and most of all no exhausts. Electric vehicles are also far more energy efficient than gasoline engines and they are very quiet too.

Electric vehicles have been in continual use since the 1900s in various applications. Today these quiet vehicles with no tailpipe emissions are no longer limited to golf carts. New advances in battery technology, system integration and aerodynamics, and research and development by major vehicle manufacturers have led to the producing of electric vehicles that will play a practical role on city streets.

1.1 University of Canterbury's Electric Vehicle

The University of Canterbury has been involved in the electric vehicle development since 1974 [2]. The objective was to provide a suitable laboratory test bed by which to demonstrate the drive system, control system and various other systems. The University's first electric vehicle development, the "Mark I" was completed in 1976 [3]. Although the vehicle had a limited range of only 40km on one charge, it had demonstrated that an electric powered vehicle could be viable for operation in urban driving conditions.

In the early 1980's a second electric vehicle, the "Mark II" was completed in conjunction with the Mechanical Engineering department. The vehicle was based on a 1962 Austin A40 Farina as shown in Figure 1.2. Major body is undertaken to lower the car's aerodynamic drag. With its much improved drag co-efficient over the Mark I, the range of the Mark II is improved to 60km on one charge at a constant 65km/h speed with a top speed of up to 80km/h [4]. As of February 2000, the Mark II had been registered for road use for over 18 months and has clocked up more than 2000 km. The car is now sporting new University logo and a new paint job.

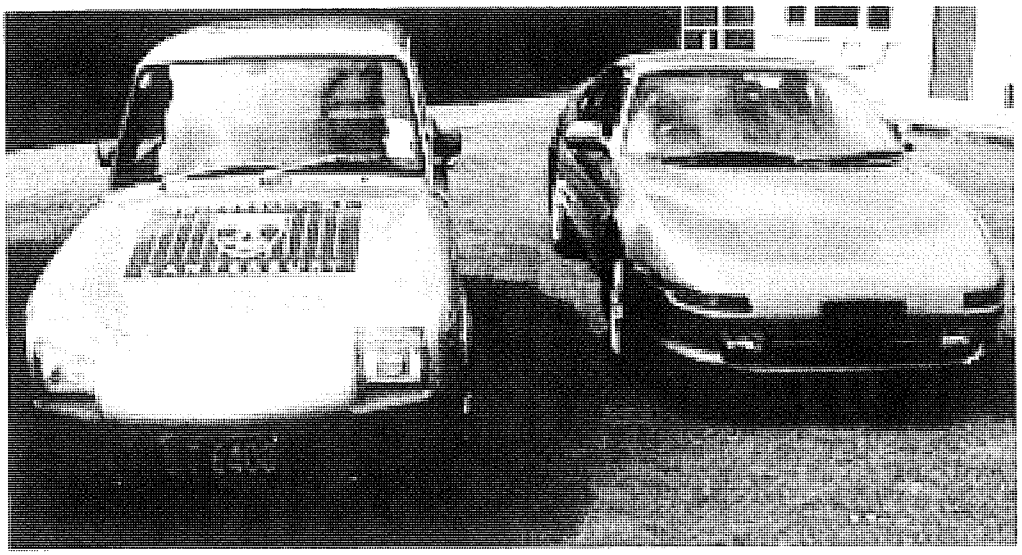


Figure 1.2 University of Canterbury's Mark II (left) and MR2 (right) electric car [5]

The University of Canterbury has purchased a Toyota "MR2" for use as the next version in the electric car research programme. The car as shown in Figure 1.2 was purchased in the early December 1999. Initial planning has commenced on this project by interested staff and students. It is expected to take 2 to 5 years for the conversion and the main areas to investigate are: Auxiliary system, Battery system, Drive system, and Mechanical modifications [5]. This thesis focuses on the Auxiliary system of the MR2.

Similar to the common combustion engine vehicles, electric vehicles requires power supplies and control systems to operate the 12V auxiliaries, such as lights, indicators and wipers. Nowadays, the power and control system for the auxiliaries normally use a pair of wires for supplying power and a single wire for controlling each auxiliary load in a vehicle. This results in a large number of wires in the wiring harness.

An alternative power and control system for the auxiliaries is to use a pair of control lines and a universal power connection around the vehicle. At selected points around the vehicle, small control units are placed to apply power to the desired auxiliary, depending on the current requirements. This reduces the number of wires in the harness and makes it easier to isolate specific circuits. This power and control system is named the "Power Distribution Network" and it is implemented by using multiple power converters (the Auxiliary Power System) and a differential control system (the CAN Control System).

1.2 Thesis Outline

This thesis looks at the development of the Power Distribution Network for the electric vehicle, MR2 application. The Power Distribution Network is developed to control and supply power to the auxiliary loads in the vehicle. The main objectives are to demonstrate the operation of the Power Distribution Network, and to investigate whether the efficiency of operation is able to provide fast and reliable control signals to the auxiliary loads.

Chapter 2 of this thesis introduces the Auxiliary Power System and the CAN Control System of the Power Distribution Network including their operating principles. Chapter 3 presents the design of the Auxiliary Power System and its control and protection circuitries are detailed in Chapter 4. In Chapter 5, the design of the CAN Control System is presented and Chapter 6 describes the components and construction of a prototype CAN Control System. Experimental testing is then undertaken to examine the operating characteristics of the completed Power Distribution Network prototype. The results of which are given in Chapter 7. Finally, further development and conclusions are drawn in Chapter 8.

2 POWER DISTRIBUTION NETWORK

The University of Canterbury purchased a modern Toyota MR2 sports car with the intention of converting it into an electric vehicle. This chapter presents the plan of designing the Power Distribution Network for the MR2. In the electric vehicle, the Power Distribution Network is designed to control and supplies power to the auxiliaries such as power windows, indicator lights and power steering.

In an electric vehicle, the electrical power is the only source that keeps the vehicle moving. Unlike the common combustion engine vehicle, where the battery can be charged by an alternator while it is travelling. All the major components/parts in an electric vehicle, for instance, the motor, power inverters and all the auxiliary loads, use the limited power that is supplied by the batteries. Therefore in an electric vehicle, energy efficiency is extremely important and the Power Distribution Network must be designed for high efficiency.

Figure 2.1 shows the Power Distribution Network that is planned for this electric vehicle. The nominal battery voltage of the electric vehicle will be 300Vdc. To provide power to the auxiliary loads, the voltage is stepped down to 12Vdc in two stages. The Battery Voltage Conversion Station block produces a 48Vdc intermediate distribution bus, while the Auxiliary Power System block produces the 12Vdc.

The 48Vdc voltage distribution network is considered more efficient than the usual 12Vdc distribution system because of the following reasons. First, it had a higher voltage which resulted in a lower current that could be withdrawn to deliver the same amount of power to the loads when compared with the usual 12Vdc distribution system.

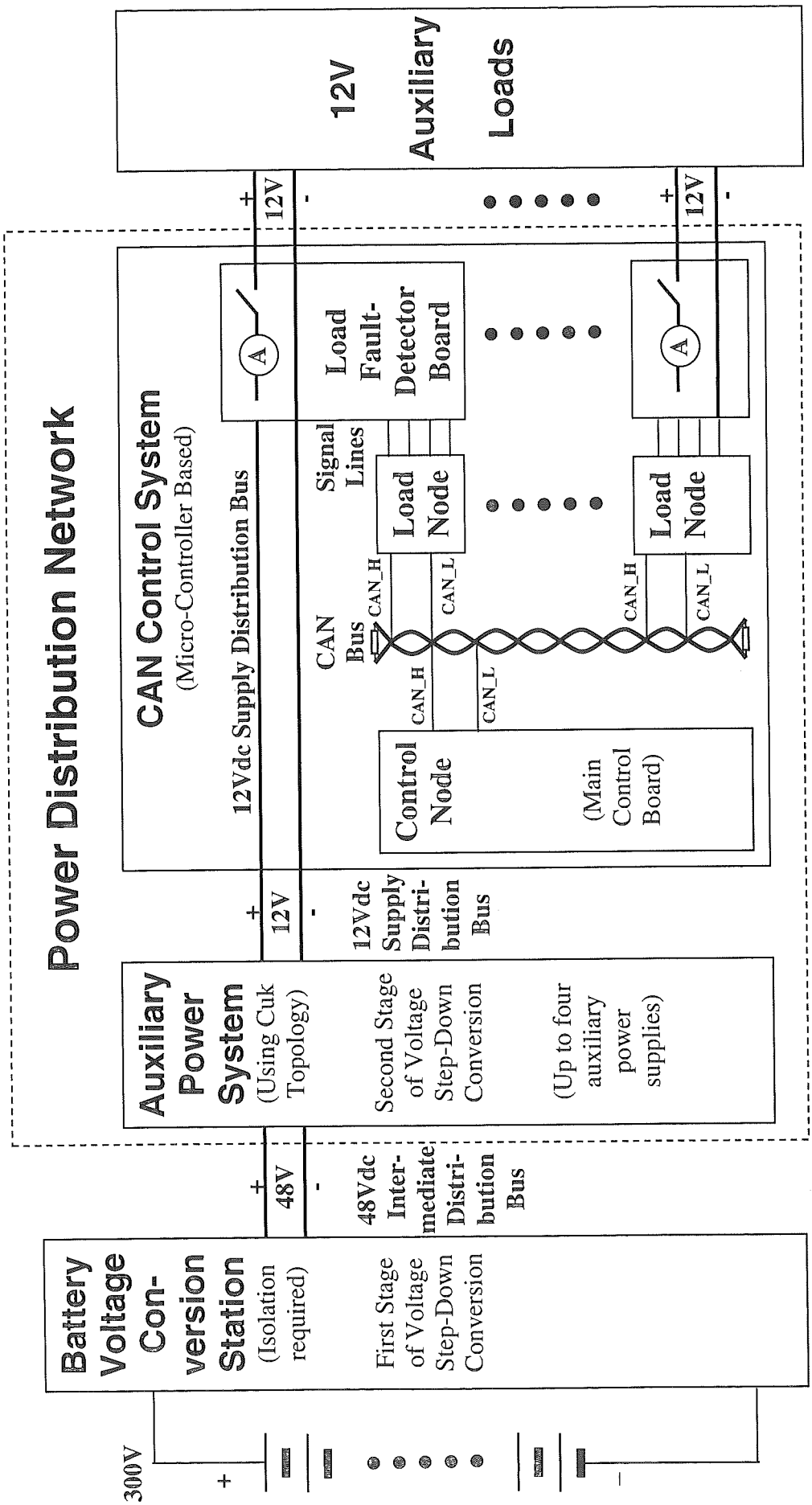


Figure 2.1 Power Distribution Network Overall Diagram

This is a decreased power loss in the wires due to lower current is carried. Second, with a 48Vdc voltage bus, the weight and space required for the wiring could be reduced since smaller and lighter wires could be used in the vehicle. While carrying on with this concept, a 300Vdc distribution network may be more efficient, it is not considered an option because of safety issues. As illustrated in Figure 2.1, the Power Distribution Network consisted of two major systems:

1. The Auxiliary Power System, it is designed to provide power to the loads using up to four distributed auxiliary power supplies.
2. The CAN (Controller Area Network) Control System, it is designed to provide control to the loads using the standardised Controller Area Network Bus (so-called the CAN Bus) topology. The system has another function of monitoring the status of the loads.

All further details about these two systems will be presented later in the chapter.

2.1 Auxiliary Load Current Investigation

The power requirements of loads that are to be retained in the petrol-electric conversion need to be determined. The power rating is one of the factors that helped determine the type of dc-dc converters that will be used. Current consumption of the loads is important for an electric vehicle as it determines the power rating, number and placement of the dc-dc converters that are going to be built for the Auxiliary Power System in the vehicle.

Auxiliary loads in the vehicle can be classified into two main categories: resistive and inductive. Resistive loads consist of lightings such as headlights, external and internal lights. Other electronics like car audio system and rear defogger are also considered as resistive loads. Inductive loads normally involve motors, such as power steering, heater fans, power window and window wipers. Table 2.1 summarises all the auxiliary loads that are in the MR2 vehicle.

Table 2.1 Auxiliary loads in the vehicle – MR2

Loads	Type	Description
Headlights	Resistive	Two light bulbs inside each headlight for high beam and low beam sight.
External Lights	Resistive	Included red taillights (in dim or brake situation), fog lights (not working), reverse lights and hazard (signal) lights.
Internal Lights	Resistive	Included door lights, reading lights and boot light.
Power Steering	Inductive	Included a motor for hydraulic pressure.
Fan Heater	Resistive	Heating elements.
	Inductive	Included air conditioning system (not working) and fans.
Other Electronics	Resistive	Included car audio system (CD/Tape/Radio player and speakers), real defogger and combination meters.
	Inductive	Included power windows, window wipers, central locking, radio aerial winder, wing mirror and horn.

Measurements are made on the MR2's auxiliary loads. This is harder than first expected. The wiring system is hidden away and difficult to reach. Further measurements may need to be made once the conversion of the car takes place. A multi-meter is used to get some purely average DC current readings from the fuse boxes. Fuses are removed and the probes of the multi-meter are used to complete the circuit and the currents are measured in this way. This method is not acceptable for all measurements due to the fluctuations in current and the type of load. An oscilloscope is used to capture the actual current waveforms. This is done with the addition of a loop of wire into the fuse holder and a CT clamp used in conjunction with the oscilloscope. These waveforms are used to gather detailed information about the auxiliary loads of the Toyota MR2 and the loading requirements for the future Power Distribution Network. Thus, they are all recorded and the magnitude of currents are measured and tabulated for easy reference in Appendix A1.

Transient current peaks occurred in the measurements and they exist because loads are switched in and out. Figure 2.2 shows an example of measured red tail-lights current waveform with a dim setting. It appeared that when the light is turned on, the current rises up immediately to an enormous peak value of 26.4A. Although the current is then stabilised relatively quickly down to 5A within 75ms, the amount of the high current peak may still load down the auxiliary power supplies and cause them to drop out or shut down.

Table 2.2 shows the summarised current consumption of the auxiliary loads in the vehicle. It is seen that the headlights, other external lights and power steering consumed the major amount of power. It also shows that the magnitude of current peaks is extremely large. They are about twice as large as the average current of the loads. These current peaks resulted in a significant power loss and this can not be afforded by the supply batteries because of the limited energy stored in it. Therefore, efficient use of the power becomes a major concern in the design.

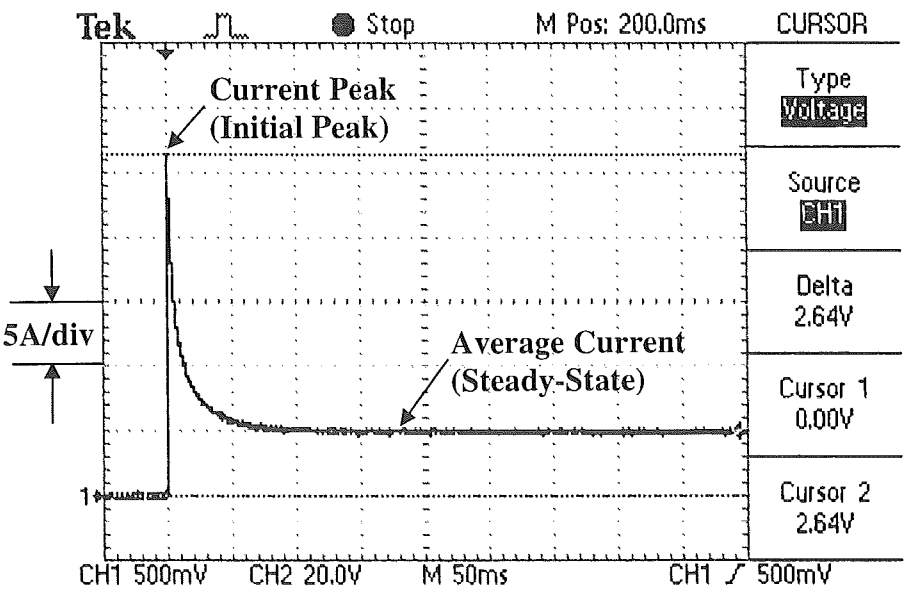


Figure 2.2 Current waveform of red taillights in dim situation (Current scale: 5A/div; Time scale: 50ms/div)

Table 2.2 Maximum current demand of the loads

Loads	Maximum Current Consumed	
	Peak Current (Initial peak)	Average Current (Steady-State)
Headlights	27.4A	15A
External Lights	53.8A	31A
Internal Lights	10.3A	2.1A
Power Steering	49.2A	25A
Fan Heater	25.6A	11.8A
All Other Electronics (see Table 2.1 for descriptions)	42.1A	20A
Total	208.4A	104.9A

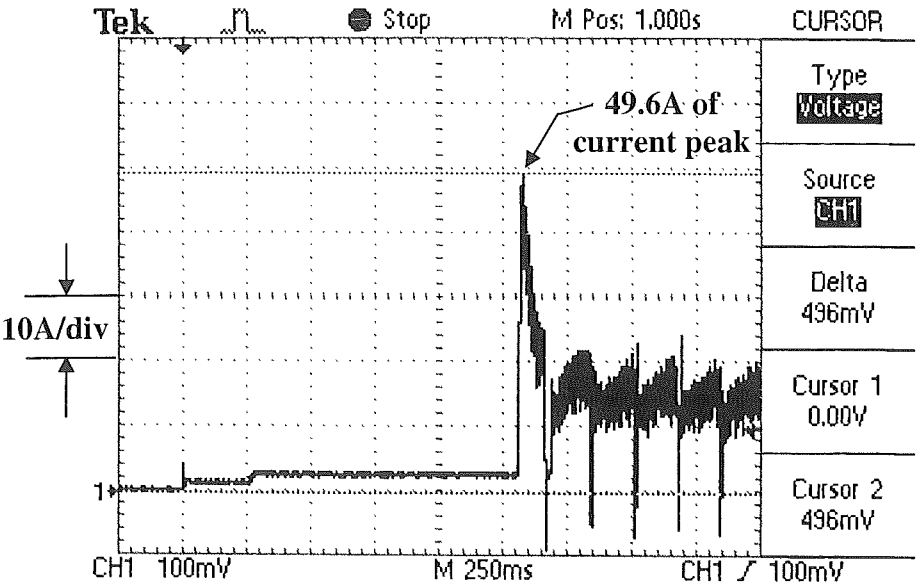
2.2 Efficiency Improvement Method

Current peaks have to be reduced as they lead to a significant power loss in the system. In order to minimise the peaks, an experiment to check performance under current limiting conditions is undertaken on most of the loads. In the experiment, the auxiliary parts are tested under the condition of limiting the supply current to its average value (see Table 2.2) and seeing whether it still functions the same.

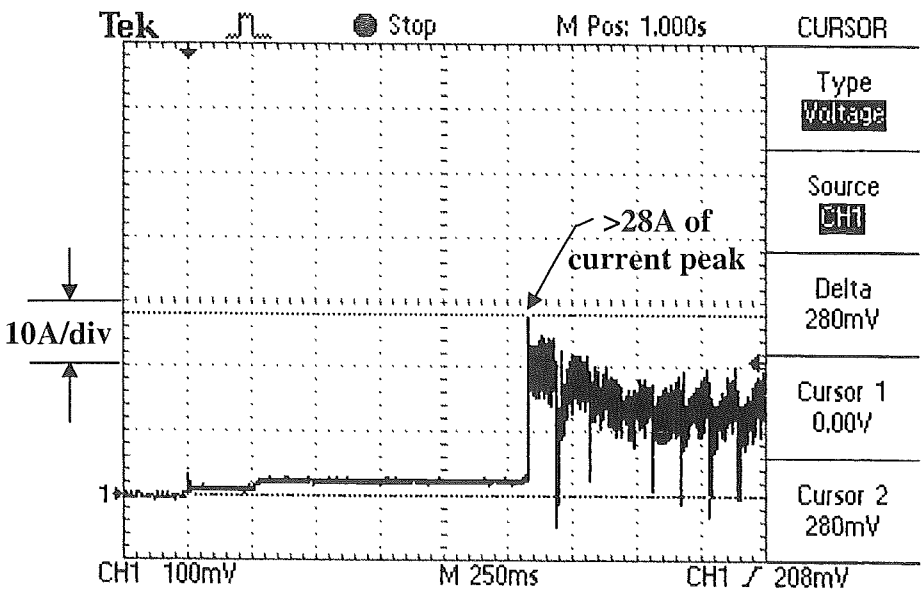
Since all the auxiliary loads are classified as being either resistive or inductive as indicated in Table 2.1, an auxiliary load from each category is picked as an example to demonstrate how the experiment is carried out. Power steering (inductive) and hazard lights (resistive) are selected due to their significant power consumption and high peak current as shown in Figure 2.3(a) and Figure 2.4(a).

In the experiment, power steering is tested using 25A as the limited current. The power steering operates successfully at current levels less than 25A. Peak current limiting only occurs when the steering operates in hard lock. In hard lock no power steering boot is required so the current limit to 25A does not affect the steering operation. Figure 2.3(b) shows the current waveform of the power steering under the current limiting test of 25A. The graph revealed that the current peak has been reduced significantly from the original 49.6A down to less than 28A. The use of current limiting result in reduced energy consumption from the supply batteries.

Figure 2.4(b) shows the current waveform of the hazard lights captured when 8A of current limiting is applied. It appeared that the current has been reduced successfully down to 16A from the original current peak of 24.4A as shown in Figure 2.4(a). Note that with the current limiting on the hazard lights, a delay time of 380ms, due to current limiting exists before the hazard lights start to flash. This is not considered as a problem because 380ms is not a significant delay time which may be noticed by people.

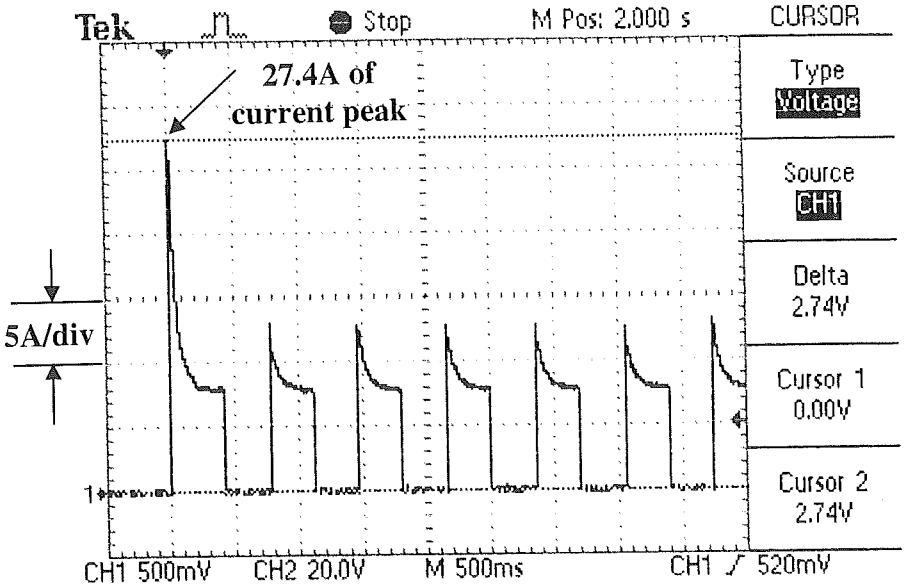


(a)

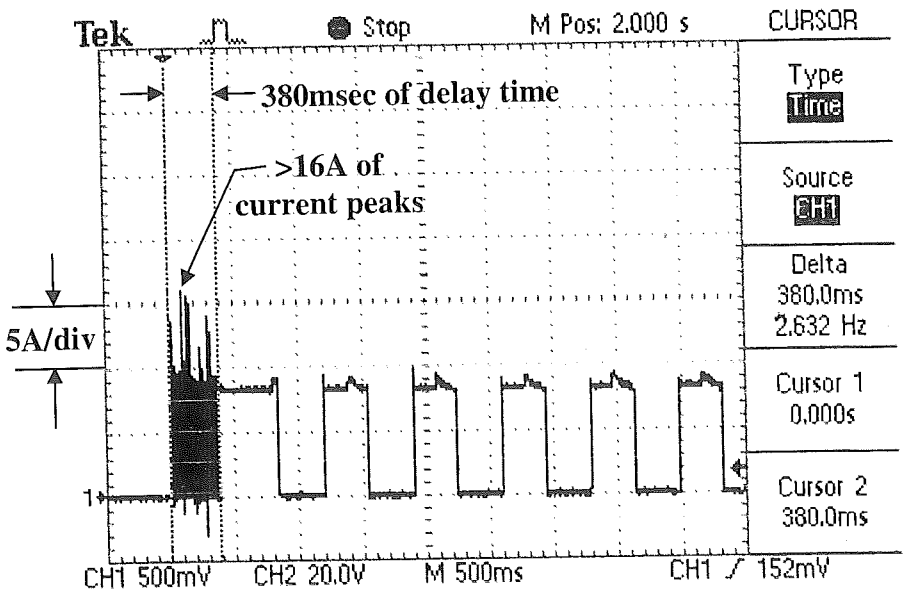


(b)

Figure 2.3 (a) Current waveform of the power steering with no current limiting applied. (Current scale: 10A/div; Time scale: 250ms/div)
(b) Current waveform of the power steering under the 25A of current limiting test. (Current scale: 10A/div; Time scale: 250ms/div)



(a)



(b)

Figure 2.4 (a) Current waveform of the hazard lights with no current limiting applied. (Current scale: 5A/div; Time scale: 500ms/div)

(b) Current waveform of the hazard lights under the 8A of current limiting test. (Current scale: 5A/div; Time scale: 500ms/div)

All other auxiliary parts under the test are found to operate properly in the current limit tests, thus, the power ratings of the converters are decided based on the average current values of the loads. Also, the current limiting capability is required to be added to these converters in order to reduce the peak current demand of the auxiliary loads.

Another way of reducing power consumption in the car is to replace all the external and internal light bulbs in the vehicle, except headlights, with high intensity LEDs. Preliminary calculations are carried out to show that the current consumption of these loads can be decreased significantly if LEDs are used. For example, the current consumption of external lights could be reduced from 31A to 10A by replacing all the external lights by LEDs. Table 2.3 summarised current consumption of the external and internal light bulbs in the vehicle after the high intensity LEDs are applied.

Table 2.3 Power consumption of the vehicle lights before and after the replacement of high intensity LEDs

Lights	Current Required in normal light bulbs	Current Required in high intensity LEDs
External Lights (Excluded Headlights)	31A	10A
Internal Lights	2.1A	0.7A

2.3 Power Supply Placement in the MR2

Using the information provided by the Table 2.2 and Table 2.3, a decision is made to use up to four converters with output current ratings that ranged from 20 to 30A. Specifically, as the average current consumed in power steering is around 25A, one converter with a 30A output is placed nearby it to supply the power. Another 30A power supply is going to be located in front of the vehicle and supplied the headlights and front external lights. A 20A converter is placed behind the driver's seat to supply the internal lights and heater fans. Finally, the power of all other electronic equipments and the external taillights is supplied by a 30A power supply inside the boot of the vehicle. Figure 2.5 illustrates the placement of the four power supplies used in the vehicle.

As an initial step in the project, a dc-dc converter with a 30A output current has been built and deign procedures are presented in the following chapters.

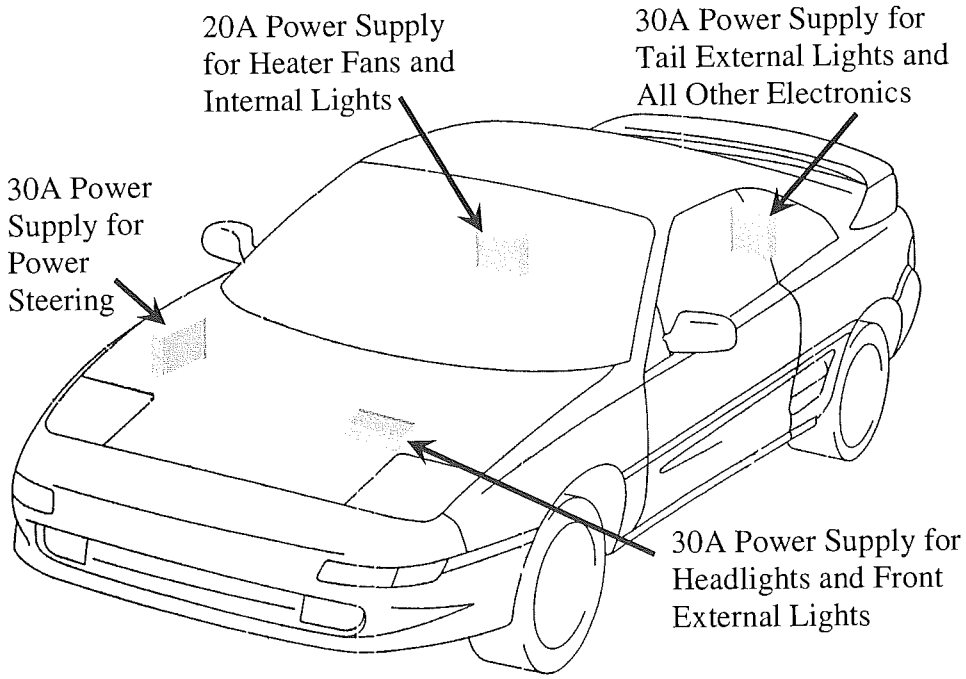


Figure 2.5 Power supply placements in MR2

2.4 Auxiliary Power System

The Auxiliary Power System is designed to provide power to the 12Vdc auxiliary loads with various current ratings. The power received from the 48Vdc intermediate bus could be stepped-down to 12Vdc either by using a single power supply with large power rating or by using number of lower power rating power supplies. Since efficiency is the major concern of the research, smaller power rating power supplies will be built because of lower current that could be withdrawn and the reduction in the power loss in the wires. Another advantage of this configuration is that smaller wire and wire loops can be formed in the vehicle to reduce the space required and radiated RFI (Radio Frequency Interferences).

The Auxiliary Power System consisted of four power supplies with differences in the power consumption due to the type of individual auxiliary load. These power supplies are dc-dc step-down converters with no transformer isolation as isolation is already involved in the Battery Voltage Conversion Station block as shown in Figure 2.1.

To design the Auxiliary Power System, all the possible topologies of dc-dc converters are required to be researched. Since efficiency is the major concern in the application, a high efficiency converter is required to be selected as the most appropriate topology to supply the various distributed auxiliary loads. More information on the converter's topology research and its design are presented in Chapter 3. Chapter 4 described the control circuitry and construction of the converter.

2.5 CAN (Controller Area Network) Control System

The CAN Control System is developed to control the auxiliary loads such that the auxiliary loads could be turned on and off using logic level MOSFET switches. It is also designed to monitor the status of the auxiliary loads which provide extra protection on both the network and loads.

The CAN Control System consisted of various micro-controller and CAN protocol controller based circuit boards (so-called CAN nodes). All the CAN nodes are connected to a twisted pair of wires (so-called the CAN Bus) to transmit and receive messages. The reason of selecting CAN is because it reduced the wiring harness weight and complexity in the vehicle. The design and construction of the CAN Control System are presented in the Chapter 5 and Chapter 6.

2.6 Summary

The Power Distribution Network is presented and can be used in an electric vehicle application. It is divided into two major systems: the Auxiliary Power System, which is in charged of converting and delivering power to the loads, and the CAN Control System, that is used to control and monitor the auxiliary loads.

The 300Vdc nominal battery voltage is converted to 48V intermediate voltage. This configuration is considered more efficient than the usual 12Vdc distribution system since smaller and lighter wires can be used to carry same amount of power. Current consumption investigations are made on the MR2's auxiliary loads. Transient current peaks occurr when switch loads are switched in and out. These may load down the auxiliary power supplies and cause them to drop out or shut down. In order to use the limited battery energy more efficiently, current limiting capability is added to the converters with the intention of reducing the peak current demand of the auxiliary loads. Additionally, high intensity LEDs could be used to replace the existing light bulbs to further improve efficiency.

The type of loads determines the power rating, number and placement of the auxiliary dc-dc converters that form the Auxiliary Power System in the vehicle. The converter's topology research and its design are presented in the next chapter, Chapter 3.

3 CONVERTER DESIGN

In order to implement the Power Distribution Network as outlined in chapter 2, the Power Distribution Network must provide both power and control to all the auxiliary loads in the vehicle. In order to provide the power, an Auxiliary Power System is required to convert voltage from 48V-to-12Vdc using up to four distributed auxiliary power supplies as mentioned in Section 2.3. The required power ratings of these power supplies are in the range of 240Watts to 360Watts. There are three major dc-to-dc power supply technologies that can be considered to implement the auxiliary power supplies and they are:

1. Linear regulators.
2. Pulse Width Modulated (PWM) switching power supplies.
3. Resonant/quasi-resonant switching power supplies.

Linear regulators are used predominantly in ground-based equipment where the generation of heat and low efficiency are not of major concern. Each linear regulator has an average efficiency of between 35 and 50 percent and the loss is dissipated as heat. Since energy efficiency is the major requirement in the electric vehicle application, linear regulators have not been considered as an appropriate topology for the auxiliary power supplies.

Topologies of PWM switching power supplies are more suitable for the auxiliary power supplies due to their higher efficiency (between 70 and 85 percent) and more flexible use than linear regulators. One commonly finds them used within aircraft, automotive products and generally those applications where high efficiency is required. Their weight is much less than that of linear regulators since they required less heatsinking for the same output power ratings.

Resonant/quasi-resonant switching power supplies can be built with a smaller size and lighter weight when compared to PWM switching power supplies for the same output power ratings. This is because of the resonant/quasi-resonant switching power supplies are normally operated with higher switching frequencies (> 200 kHz), which decreases the size of the magnetic components. They are also suitable for applications that required high output power ratings (> 1kW). Since the auxiliary power supplies have a power rating of less than 500W and does not required to be operated with a very high switching frequency, resonant/quasi-resonant switching power supplies have not been selected although they may offer higher efficiencies.

There are numbers of possible converter topologies using PWM switching technology. Transformer isolation was not required in the auxiliary power supplies since isolation exists in the Battery Voltage Conversion Station (see Figure 2.1). However, transformers can still be used to provide step down ratios. Topologies of transformer isolated PWM converters are investigated; these include Forward, Flyback, Push-pull, Half-bridge and Full-bridge converters.

A Full-bridge converter is not considered due to it being more applicable to higher output power ratings (>1kW). Push-pull and Half-bridge converters are not selected, because the power supply is required to be reliable and controlled by single switch if possible. Since the Flyback converter has a discontinuous input current, and with a small duty ratio will draw a high peak current (around 30Ap-p for the design) from the output of the Battery Voltage Conversion Station. This poses a significant difficulty for the conversion design; therefore, the Flyback converter is not a suitable choice. A Forward converter is not selected due to its magnetic complexity and extra windings are needed which lead to an increase of the converter's overall size.

Although that the transformer isolated converters may develop a slightly higher overall efficiency, when compared to the non-isolated PWM converters, they are not suitable for the design due to their characteristics as mentioned above and the transformer isolation was not required. Topologies of non-isolated PWM dc-dc converters are therefore selected to implement the auxiliary power supplies.

A standard approach of using non-isolated PWM technology to convert 48-to-12Vdc is to use a buck converter with an output filter. However, due to the discontinuous input current characteristic of buck converters, the topology is found not to be acceptable.

The Cuk converter topology is selected because it has both continuous input and output current with a low current ripple. Since the input and output inductor currents are essentially constant in steady state, the switching current is confined entirely within the converter, in the transistor-coupling capacitor-diode loop. With careful layout, this loop can be made physically small, which will reduce the radiated RFI and improves the efficiency. In addition, the voltage and current waveforms in this converter are particularly clean, with very little ringing or overshoot, and therefore very little snubbing is required [6].

As an initial stage in designing the auxiliary power supplies using Cuk converter topology, a 360W Cuk converter is designed and implemented as the power supply for the power steering in the vehicle.

The following sections of the chapter present the design of the 360W Cuk converter. This includes a brief description of the Cuk topology, followed by the design objectives and specifications of the converter implementation. Switching frequency selection is found to be the key parameter to gain the best compromise between the converter's size and efficiency. Therefore, preliminary calculations and PSPICE simulations are made to estimate the component loss in the converter. Then a simulated Cuk Converter's efficiency graph, over a range of switching frequencies, is derived to select the switching frequency of the converter. After that, inductor design procedures are presented and followed by the reasons for component selection. Finally, a summary is attached in the end of the chapter to summarise the design of the 360W auxiliary power supply using Cuk converter topology.

3.1 Cuk Converter Topology

Figure 3.1 illustrates the schematic Cuk converter and Table 3.1 shows the theoretical formulae derived from the operation [7]. Referring to the Figure 3.1, when switch T1 is open, the diode D1 is forward biased and capacitance C1 is charging in

positive direction through inductance L1. Consequently, during the interval when T1 is closed, D1 is reverse biased and C1 discharge through L2 and load RL, thus charging the output capacitance C2 to a negative voltage. Finally, to close the complete cycle, when T1 is opened again, D1 conducts again and the capacitance C1 is charged positively using the stored energy in L2. [8].

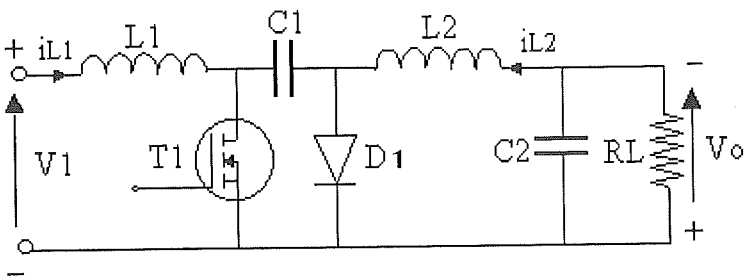


Figure 3.1 Cuk converter

Table 3.1 Equations of Cuk converter [7]

Vo/V_1	$-\frac{D}{1-D}$
$\Delta Vo/V_1$	$\frac{1-D}{8 \cdot L \cdot C \cdot f^2}$
L (Where D = Duty ratio and f = Switching frequency)	$L1 = \frac{V1 \cdot D}{f \cdot \frac{diL1}{dt}}$ $L2 = \frac{V1 \cdot D}{f \cdot \frac{diL2}{dt}}$
C (Where $\frac{dV}{dt}$ = Rate of change of voltage)	$C1 = \frac{Vo \cdot D}{RL \cdot \frac{dvC1}{dt} \cdot f}$ $C2 = \frac{Vo \cdot (1-D)}{8 \cdot \frac{dVo}{dt} \cdot L2 \cdot f^2}$

3.2 Design Objectives

In the auxiliary loads investigation carried out in Chapter 2, the normal functional voltage for auxiliaries in the vehicle is found between 11V to 13Vdc. Therefore, the output voltage supplied from the Cuk converter is specified as 12Vdc ±10%, which is inside the normal operational range of the auxiliary loads. Current ripple is not the main factor that influences the performance of the auxiliary loads in the vehicle, a maximum current ripple of 4.2Ap-p is found during the investigation. This specified the output current rating of the Cuk converter of 30A±10%. Table 3.2 tabulated the specifications of the Cuk converter design.

Table 3.2 Design requirement

Input Voltage	48Vdc
Output Voltage	12Vdc ± 10%
Output Current	30A ± 10%
Maximum Output Voltage Ripple	1.2V peak-to-peak (≤10% of 12V)
Maximum Current Ripple of Inductance L2	3A peak-to-peak (≤10% of 30A)
Minimum Efficiency	Preferably > 85%
Input Protection	Reverse voltage
Output Protection	Over voltage, Over current, Short circuit

The output voltage and current ripple values are set to ± 10% to match the tolerance of the output voltage and current. According to the paper [8], with a proper design and PCB layout, Cuk converters are be able to obtain a high efficiency of 90 percent. Thus, the efficiency of the design is preferably to be greater than 85 percent. Since the Cuk converter is required to be robust and reliable, input and output protection is required for the converters. All the details on this protection are presented in the Chapter 4.

3.3 Component Calculation

In order to design a converter, the switching frequency is required to be set initially since it determines the efficiency and size of the converter. For hard switching topologies, the switching frequency is normally in the range of 25~200 KHz. At the various switching frequencies, preliminary calculations on component sizes are made based on each individual switching frequency of 25, 50, 100, 150 and 200 kHz. Theoretical equations listed in Table 3.1 are used to carry out the component calculations.

To initial size the converter a switching frequency of 100 kHz is selected as being middle of the range. It also provides a good compromise between the efficiency and component sizes. The design specification listed in Table 3.2 is applied and the calculation results at the 100 kHz switching frequency is tabulated in Table 3.3.

Table 3.3 Calculated component sizes at the switching frequency of 100 kHz.

Duty Ratio, D	33%
Inductance, L1	211.2μH
L2	52.8μH
Capacitance, C1	33μF
C2	15.86μF
Load RL	0.4Ω

An efficiency of 100 percent is assumed for the calculations and the component values listed in Table 3.3 are the exact values derived. However, during the implementation of the converter, the assumption is proved not to be completely valid and modification to the component sizes is required. These will be discussed later in Section 3.7 and 3.8.

All the calculation results at the various switching frequencies are included in Appendix B1. Converter simulations are carried out based on the results and they are presented in the following section.

3.4 Converter Simulation

PSpice is used to simulate the converter at the discrete switching frequencies from 25 kHz to 200 kHz. The simulation results are included in Appendix B2. Since the switching frequency of 100 kHz is presumed to provide a good compromise between the efficiency and component size as mentioned in Section 3.2, the calculation results listed in Table 3.3 are used in the simulation.

Figure 3.2 shows the simulation model of the 360W Cuk converter in Pspice. The 48V DC source is assumed to be constant and ripple free and the switching devices, s1 and d are assumed to be ideal. The switching devices, s1 and d are also modelled with a MOSFET model RFP40N10 (40A/100V) and a diode model 30CPQ100 (30A/100V, Schottky). These switching device models are chosen as they have the closest ratings to the results of the theoretical calculations [9]. Further details on the selecting reasons will be discussed in Section 3.8.

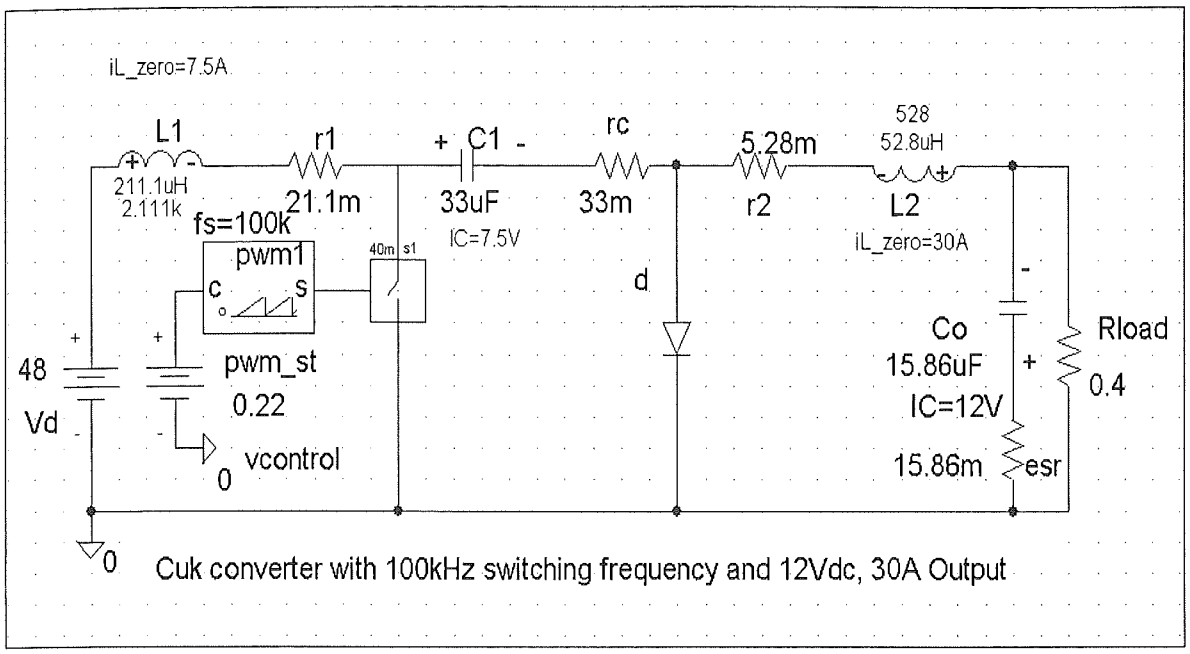


Figure 3.2 The Cuk converter simulation model

As shown in Figure 3.2, the passive components, L1, L2, C1 and C2 each simulated with a series resistance component. These series resistances present the losses of the passive components and the values are derived from the theoretical calculations [9]. All the further details in the circuit loss estimation are presented in the following section.

Figure 3.3 shows the voltage and current waveforms generated from the simulation at the switching frequency of 100 kHz. The waveforms are captured when the steady-state is reached and lasted for two switching periods.

A voltage ripple of 140mVp-p (2%) is obtained at the output of the converter as shown in Figure 3.3(a). Figure 3.3(b) and (c) show the inductor L1 and L2 current waveforms respectively. In inductor L2 the current ripple is 2Ap-p (6.7%). A large amount of pulsed current, 42Ap-p, flows through the switch and capacitor C1 on each PWM cycle as shown in Figure 3.3(d) and (e) respectively. These waveforms showed the voltage and current ripple requirement set in Table 3.2 has been meet in the simulation.

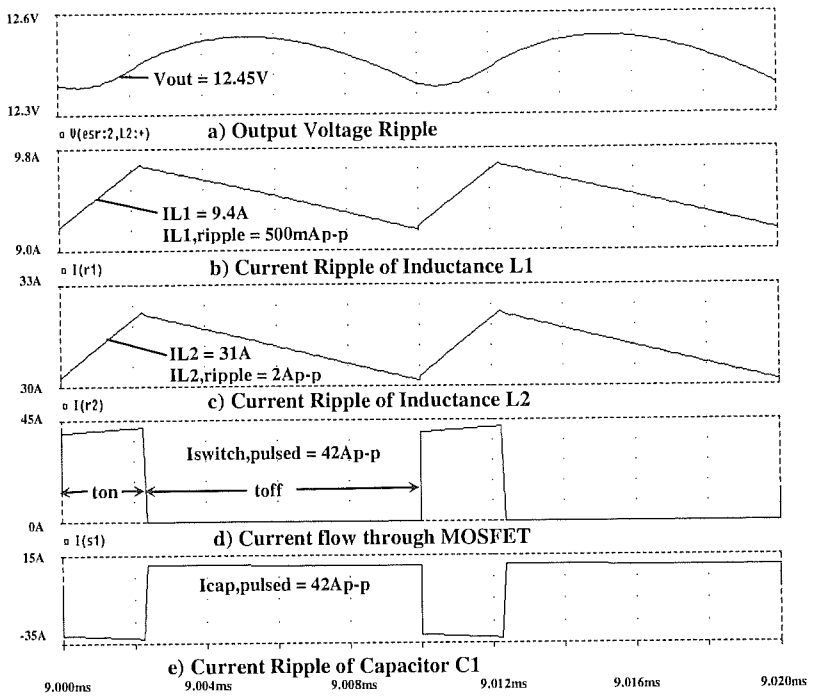


Figure 3.3 Simulation waveforms

3.5 Circuit Losses Estimation

At the various switching frequencies, theoretical calculations are made to determine the component values and their losses [9]. Figure 3.4 illustrates the estimated component losses of the converter over a range of switching frequencies.

In the hard-switching PWM converter, the major power losses are the conduction and switching losses in the switching devices. Conduction losses occur because of the voltage drop across the device and the current flowing through the device occur simultaneously during conduction. Switching losses are incurred by the simultaneous presence of voltage and current during switching. These losses can be calculated using simplified device model.

3.5.1 Conduction Loss in the Switching Devices

For the conduction loss calculation, the device can be simplified either as a constant voltage drop in or a linear resistor. The conduction loss in MOSFETs and diodes can be calculated using the measured values of turn-on drain-to-source resistance, Ron and the forward voltage drop, V_f respectively from the data sheets. The conduction loss of a MOSFET and a diode are expressed in Equations 1 and 2 respectively.

$$Pc,_{MOSFET} = I_d^2 * Ron * D \qquad \text{(Equ. 1)}$$

$$Pc,_{diode} = V_f * I_{diode} \qquad \text{(Equ. 2)}$$

Where I_d and I_{diode} are the currents flowing through the MOSFET and diode respectively. The duty cycle, D is set to 0.22 while the calculation is carried out.

3.5.2 Switching Loss in the Switching Devices

In the hard-switching PWM converter, three components of the switching losses can be identified; MOSFET turn on losses, MOSFET turn off losses, and the losses due to diode reverse recovery. The switching losses in the hard switching PWM converter can be calculated using the measured values of switching energy from the data sheets. Data sheets normally give the measured values of turn-on, turn-off time (ton, toff) and reverse recovery time, trr. The switching losses in the hard-switching PWM converter are expressed in Equation 3 and 4 respectively as follows:

$$Ps,MOSFET = 0.5 * Vds * Id * fs * (ton+toff) \qquad \text{(Equ.3)}$$

$$Ps,diode = fs * (trr * IRM)/2 \qquad \text{(Equ.4)}$$

Where Vds is the voltage across the MOSFET, fs is the switching frequency (using 25 kHz to 200kHz) and IRM is the maximum reverse recovery current of the diode.

3.5.3 Losses in the Passive Components

The power losses in the passive components are determined by finding the product of their series resistance and the current flowing through them. Using the information

provided in Figure 3.4, PSPICE simulations are carried out again to determine the overall efficiency of the converter over a range of switching frequencies as shown in Figure 3.5.

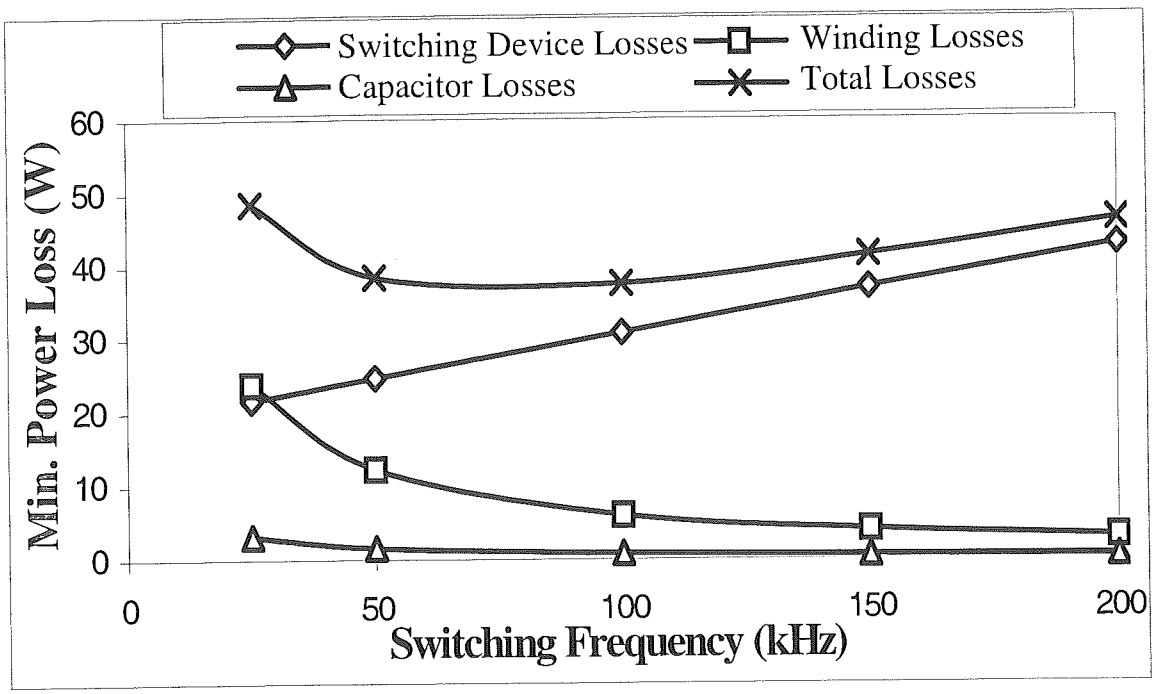


Figure 3.4 Estimated Cuk converter component loss

3.6 Switching Frequency Selection

The need for a high efficiency converter indicates a careful selection of switching frequency. This in turn has an impact on the size of the passive components. Using Figures 3.6 and 3.7 a choice of 100 kHz as the switching frequency is made. This choice is a compromise between choosing high efficiency and small components. For maximum efficiency, 75 kHz of switching frequency will be the choice. However, when looking at Figure 3.7, there is not a significant difference in efficiency between the 75 kHz and 100 kHz. But also the 100 kHz provides smaller component size. In terms of the realistic, savings in the weight and space in the vehicle is a more important factor than the marginal changes in the efficiency. Thus, to achieve a best compromise between efficiency and component size, a switching frequency of 100 kHz is used.

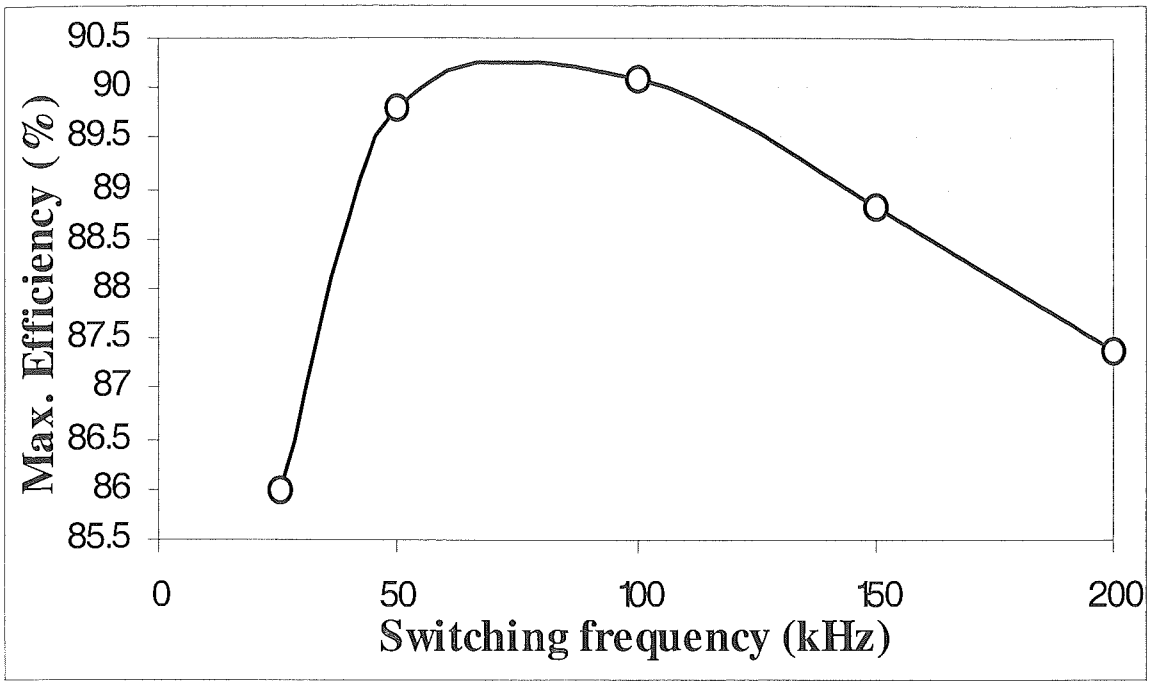


Figure 3.5 Simulated Cuk converter efficiency

3.7 Inductor Design

The design of inductor L1 is carried out before L2. The basic inductor design procedure of Billings is followed [10]. Using the simulated current ripple as shown in Figure 3.3(b) and (c) and the input parameters listed in Table 3.4, the inductors are designed.

Table 3.4 Input parameters of inductance design

Inductor	L1	L2
Inductance	211.2uH	52.8uH
Peak Current	9.6A	32A
DC Current	9.4A	31A
Max. Temp.	100°C	100°C

Inductor L1 and L2 are wound on ETD49 and ETD59 ferrite core respectively. Theoretical calculations are carried out and found that without a air-gap, both inductor cores would be saturated at the 100kHz switching frequency. Therefore an air-gap must be added in these cores to reduce the magnetic flux density. However, there is an impact on the extra windings will be needed to gain the required inductance. To achieve the best

compromise between the core size and the copper loss, an air-gap of 3mm is selected for both cores. This is verified from the tables and figures provided in the Philips Soft Ferrite data book [11]. Table 3.5 lists the results of the inductor design.

Table 3.5 Design results of input and output inductor

Inductor	L1	L2
Ferrite core	ETD49	ETD59
Air gap	3mm	3mm
No. of turns	29 turns	12 turns
Type of wire	Copper	Copper
Wire diameter	0.45mm	0.45mm
No.of parallel conductors	15 wires	35 wires
Current density	$3.6A/mm^2$	$4.5A/mm^2$
Actual inductance	213uH	52uH

3.8 Component Selections

3.8.1 Capacitor

In the Cuk converter topology, all the current transferred from input to output must pass through the capacitor C1. The converter as designed delivers high currents and this results in an enormous ripple current flowing through the capacitance C1 during each PWM period. This is confirmed by PSPICE simulation as shown in Figure 3.5(e). Due to a large ripple current of up to 40Ap-p, electrolytic capacitors are considered not to be suitable. Plastic Film type of MKT capacitors are used since they offer high AC stress properties. The capacitors that form the output filter must also be able to handle an appreciable amount of ripple current of up to 3Ap-p. Ten of 4.7μF electrolytic capacitors are placed in parallel as the output capacitors C2 to take this ripple.

3.8.2 MOSFET and Diode Switch

To maximize the efficiency of the converter, the switch is chosen to have low conduction and switching losses. An Intersil RFP40N10 N-channel MOSFET is selected to provide higher than necessary current and voltage ratings with low $R_{DS,(ON)}$ of 0.04Ω and fast on and off switching times 47ns and 62ns respectively.

Since the converter is used in a step-down application, high forward-voltage drop in the diode that will result an enormous power loss in the converter. An International Rectifier 30CPQ100 high efficiency, schottky diode is chosen that offer a low forward-voltage drop of 0.86V with the capability of handling large current and reverse blocking voltage drop of 30A and 100V respectively.

3.9 Summary

Various DC-to-DC PWM converter topologies that are suitable for implementing the Auxiliary Power System have been researched. The Cuk converter is selected because it has both continuous input and output current with high efficiency. The topology of the Cuk converter is studied and the design specification is set. PSpice simulations are then carried out to determine the characteristics of the converter under the full load condition. Theoretical calculations are made to estimate the component losses in the converter at the various switching frequencies and a switching frequency of 100 kHz was selected as the best compromise between the efficiency and the component size.

To construct the prototype of the converter, passive components and switching devices are selected according to the results from the calculations and the simulations. PWM and protection circuits are required to control and protect the converter. These control circuits for the Cuk converter are presented in the following chapter, Chapter 4.

4 CONVERTER CONTROL DESIGN

This chapter describes the circuits used to control the PWM operation of the 360W Cuk converter. The overall control design consists of implementing negative feedback compensation, current sensing and pulse-by-pulse current limiting around an analogue PWM controller IC. In addition, an automatic shut-down control circuit is also implemented to provide extra protection for against over current at the output.

There are two types of PWM control mode strategies available nowadays, the voltage mode control and the current mode control. Before starting the whole process of the control circuit design, a question is generated when looking at the design requirements listed Table 3.2. Which PWM control mode strategy is more suitable for this specific application? However, there is no single strategy which is optimum for all applications. Therefore, to answer the question as to which control strategy is the best for this specific application, one must start with knowledge of both the advantages and disadvantages of each strategy. The strategy chosen will be based on which approach closer meets the design requirements. The following sections attempts to discuss this in a consistent way for these two power supply control strategies.

4.1 Voltage Mode Control

The basic voltage mode configuration for the Cuk converter is shown in Figure 4.1. The major characteristic of this design are that there is a single voltage feedback path. The output PWM signal is generated by the LATCH after comparing the voltage error signal Ve with a constant ramp waveform (or the "sawtooth") at a fixed frequency clock. The voltage error signal, Ve, is the voltage generated after passing the output voltage, Vout, through the compensation network that consists of the error amplifier and two impedances. The

sawtooth waveform V_R is produced from the external RC circuit at the ramping frequency with respect to the clock pluses from the oscillator. In voltage mode control, current limiting must be implemented in a separate circuit. The advantages of voltage-mode control are [12]:

- A single feedback loop is easier to design and analyse.
- A low-impedance power output provides better cross-regulation for multiple output supplies.
- A large-amplitude ramp waveform provides good noise margin for a stable modulation process.

Voltage mode's disadvantages can be listed as [12]:

- Any changes in line or load must first be sensed as an output change and then corrected by the feedback loop. This usually means slow response.
- The output filter adds two poles to the control loop requiring either a dominant-pole low frequency roll-off at the error amplifier or an added zero in the compensation.
- Compensation is further complicated by the fact that the loop gain varies with input voltage.

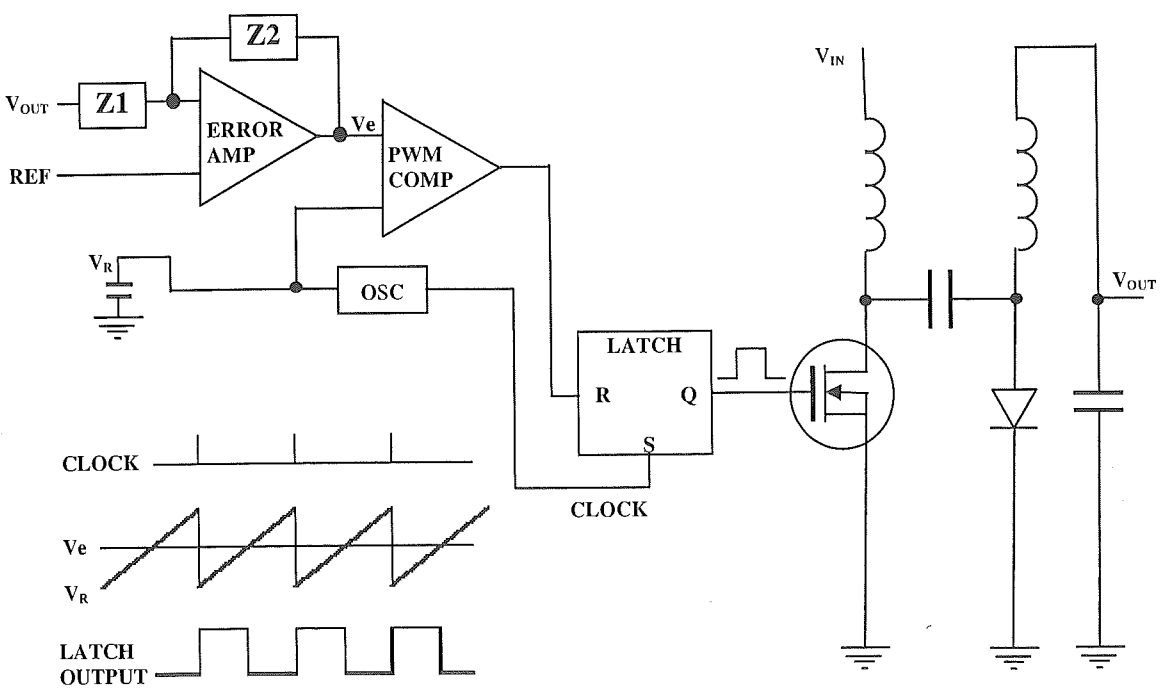


Figure 4.1 Voltage Mode Control

4.2 Current Mode Control

Another control strategy is current mode control. In this case the MOSFET current is controlled as well as the output voltage. As can be seen from the diagram of Figure 4.2, basic current mode control uses the oscillator only as a fixed frequency clock and the ramp waveform is now replaced with a signal derived from MOSFET current instead of the sawtooth as mentioned in the voltage control mode. Therefore, the PWM comparator now compares the error voltage signal, Ve, from the compensator and the MOSFET sensed current signal, $V_{SENSE,}$ from the current-to-voltage conversion resistor, R_{SENSE}. The output PWM signal is generated by the LATCH using the information provided by the PWM comparator with the respect to the fixed frequency clock pulses. The advantages which this control technique offers include the following [12]:

- Since inductor current rises with a slope determined by Vin-Vo, this waveform will respond immediately to line voltage changes, eliminating both the delayed response and gain variation with changes in input voltage.
- The Error Amplifier is now used to command an output current rather than voltage, the effect of the output inductor is minimised and the filter now offers only a single pole to the feedback loop (at least in the normal region of interest). This allows both simpler compensation and a higher gain bandwidth over a comparable voltage-mode circuit.
- Additional benefits with current-mode circuits include inherent pulse-by-pulse current limiting by merely clamping the command from the Error Amplifier, and the ease of providing load sharing when multiple power units are paralleled.

Disadvantages of current mode control are [12]:

- There are now two feedback loops, making circuit analysis more difficult.
- The control loop becomes unstable at duty cycles above 50% unless slope compensation is added.
- A particularly troublesome noise source is a current spike on the leading edge of the current signal, typically caused by transformer winding capacitance and output diode recovery current.

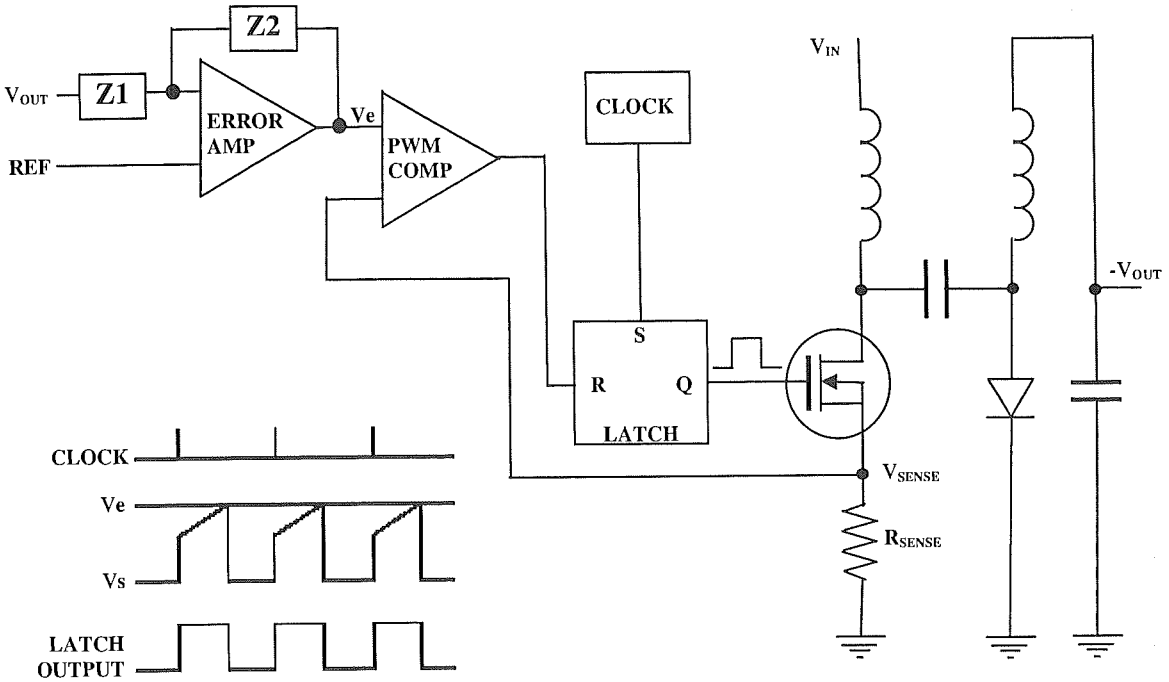


Figure 4.2 Current Mode Control

Therefore it can be concluded that current mode control overcomes the limitations of voltage mode control. Since voltage feedback compensation and pulse-by-pulse current limiting are the requirements of the project as stated in Table 3.2, the current mode control strategy is then selected. A Unitrode current mode PWM IC, UC3845 is used as the basis of the control circuit for the Cuk converter and the details of the design is discussed in the next section.

4.3 Current Mode Control Circuit Design

The current mode controller chip, UC3845 is selected since it provides the necessary features to implement a fixed frequency, current mode control Cuk converter with a minimum number of external parts. Figure 4.3 shows the simplified internal circuitry of the 8-pin IC. The UC3845 features an oscillator, a voltage reference, a high gain error amplifier, current sense comparator and a high current totem pole output ideally suited for driving a power MOSFET. Also included are protective features consisting of input and reference under-voltage lockout circuitry, and pulse-by-pulse current limiting. In addition, there is a toggle flip-flop which blanks the output on every second clock pulse, therefore

ensuring that the duty cycle never exceeds 50%. This is particularly useful for the step-down Cuk converter application. Table 3.1 lists the pin descriptions of the UC3845.

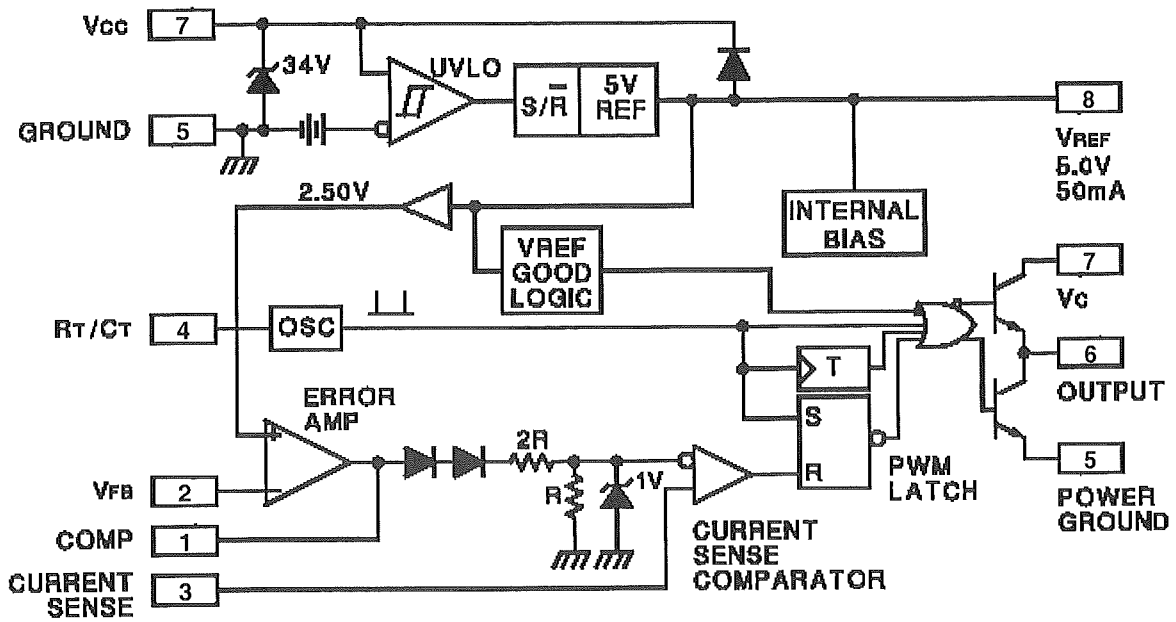


Figure 4.3 Internal circuitry of the UC3845 (Refer to Table 3.1)

Table 4.1 Pin Function Description (Refer to Figure 4.3)

Pin	Feature	Description
1	Compensation	This pin is Error Amplifier output and is made available for Type one feedback compensation loops (Section 4.4).
2	Voltage Feedback V_{FB}	This is the inverting input of the Error Amplifier. It is connected to the Cuk converter's output through the resistor divider and the OP-AMP (Section 4.4)
3	Current Sense	A voltage proportional to the MOSFET current is connected to this point. The PWM uses this information to terminate the conduction of the power MOSFET switch. (Section 4.5)
4	R_T/C_T	The oscillator frequency of 100kHz and the maximum output duty cycle of 33% are programmed by connecting a resistor R_T to the 5V V_{ref} and the capacitor C_T to the ground.
5	Ground	This pin is combined control circuitry and power ground.
6	Output	This output directly drives the gate of the power MOSFET. Peak currents of up to 1A are sourced and sunk by this pin. The output switched at one-half of the 100kHz oscillator frequency.
7	Vcc	Positive supply (+15Vdc) of the control IC.
8	V_{ref}	The 5V reference output. It provides the charging current for the capacitor C_T and through the resistor R_T.

Figure 4.4 shows the current mode control that is implemented for the Cuk converter using the UC3845 chip. There are two main control loops in the current mode control, the voltage feedback compensation and the pulse-by-pulse current limiting as shown in Figure 4.3. The designs of these two feedback loops are to be discussed in Section 4.4 and Section 4.5 respectively. Note that there is an extra input for the Automatic Shut-down Protection. It is used as an extra protection, which will shut-down the converter when over-current is sensed at the output. Section 4.6 discusses the design details of the Automatic Shut-Down Protection loop.

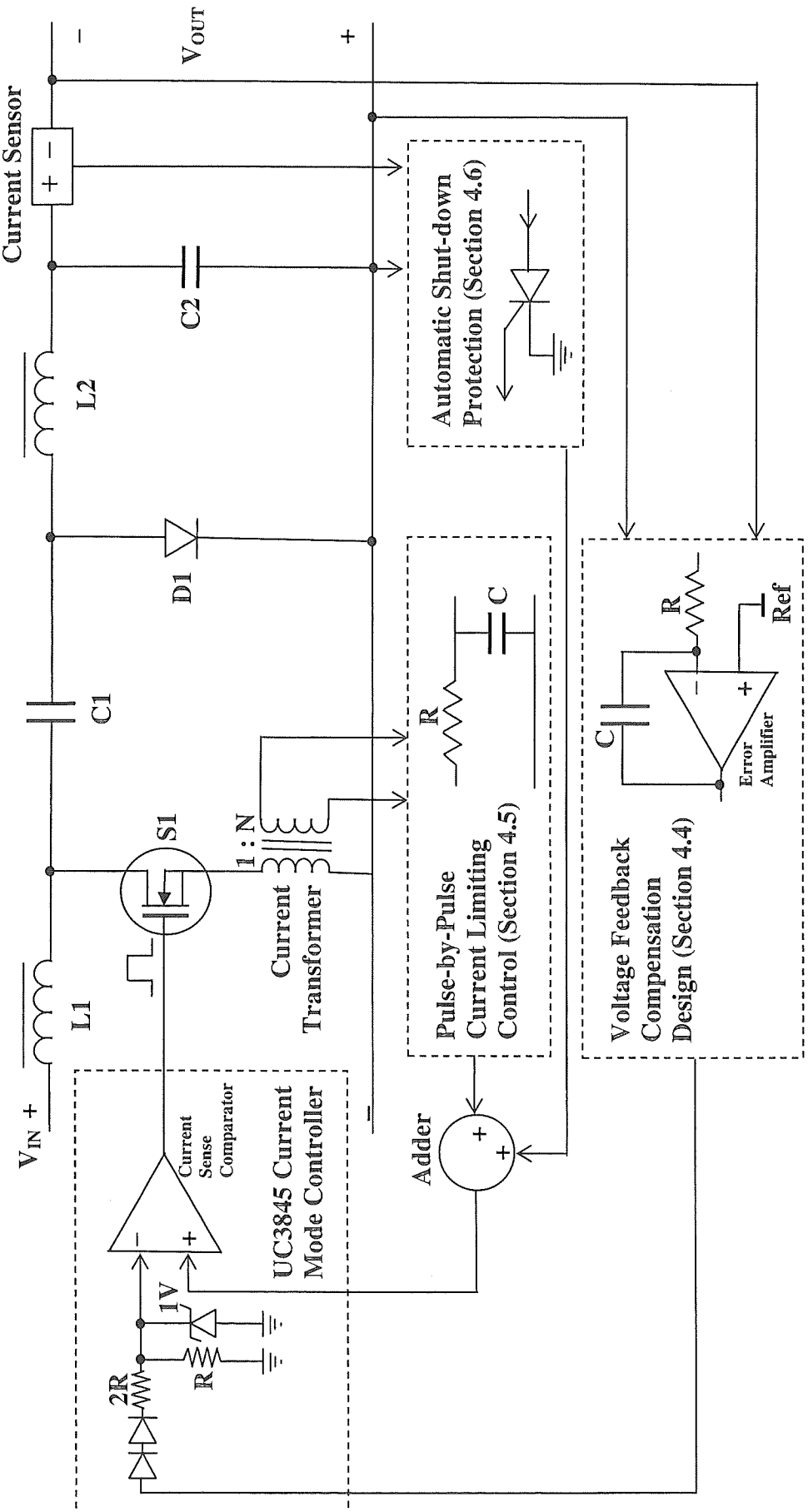


Figure 4.4 Current Mode Control Circuit for the Cuk Converter

4.4 Voltage Feedback Compensation Design

The voltage feedback compensation is designed to regulate the output at 12Vdc. This closed loop feedback also enables a fast and stable response for the Cuk converter. Figure 4.5 shows the closed loop feedback Cuk converter with the major elements separated into three parts. Block 1 is the current sense comparator, which is internally implemented in the UC3845 IC. The output of the comparator is connected to the SR LATCH, which provides the PWM output signal to drive the power MOSFET of the converter. The output of the comparator is produced by comparing the error voltage and the sensed MOSFET current.

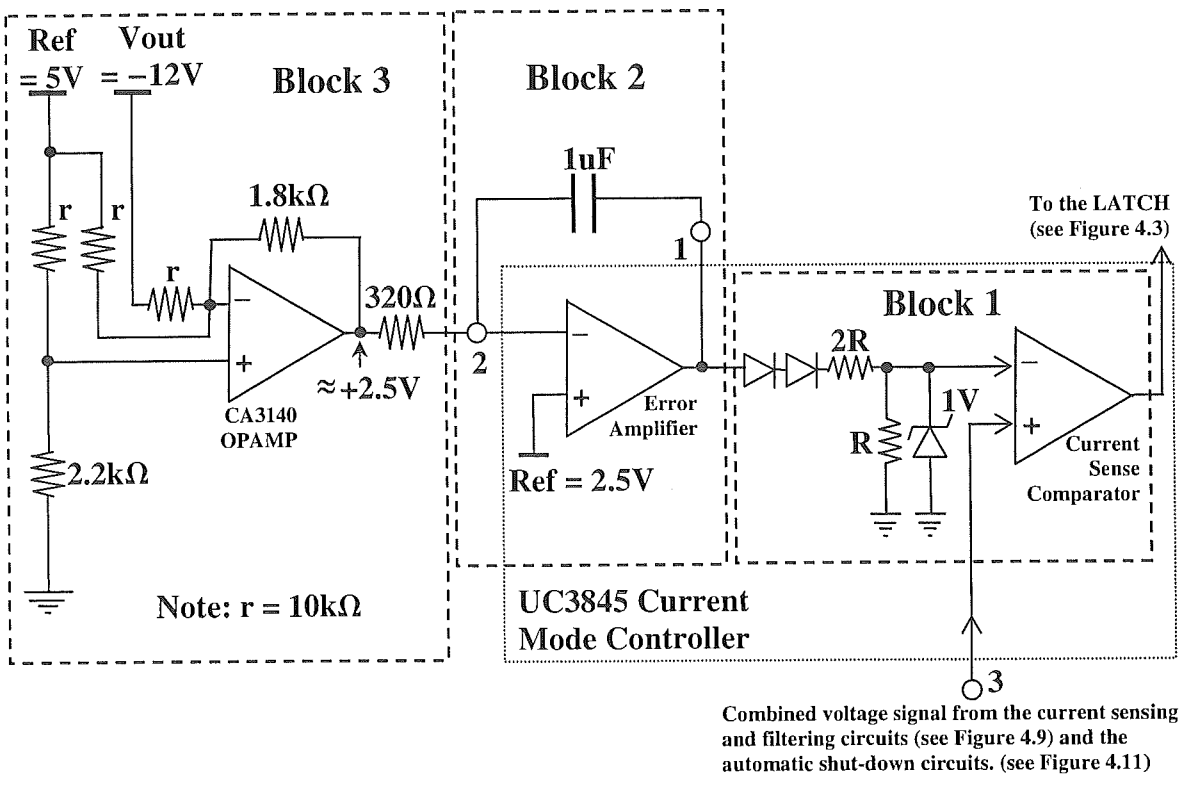


Figure 4.5 Block diagram of the voltage feedback compensation network for the Cuk converter.

Block 2 is the Type one feedback compensation. This is used to stabilise the system response by using an internal error amplifier, the internal 2.5V reference voltage and the loop compensation circuits. Due to the reverse output voltage characteristic of the Cuk converter, an additional operational amplifier circuit (as Block 3) is used to reverse the

polarity and step-down the output voltage before feeding it into the compensation circuitry in Block 2.

To design the voltage feedback compensation network in Block 2 and Block 3, the open loop system response of the converter is required. The measured open loop response of the converter is shown in Figure 4.6.

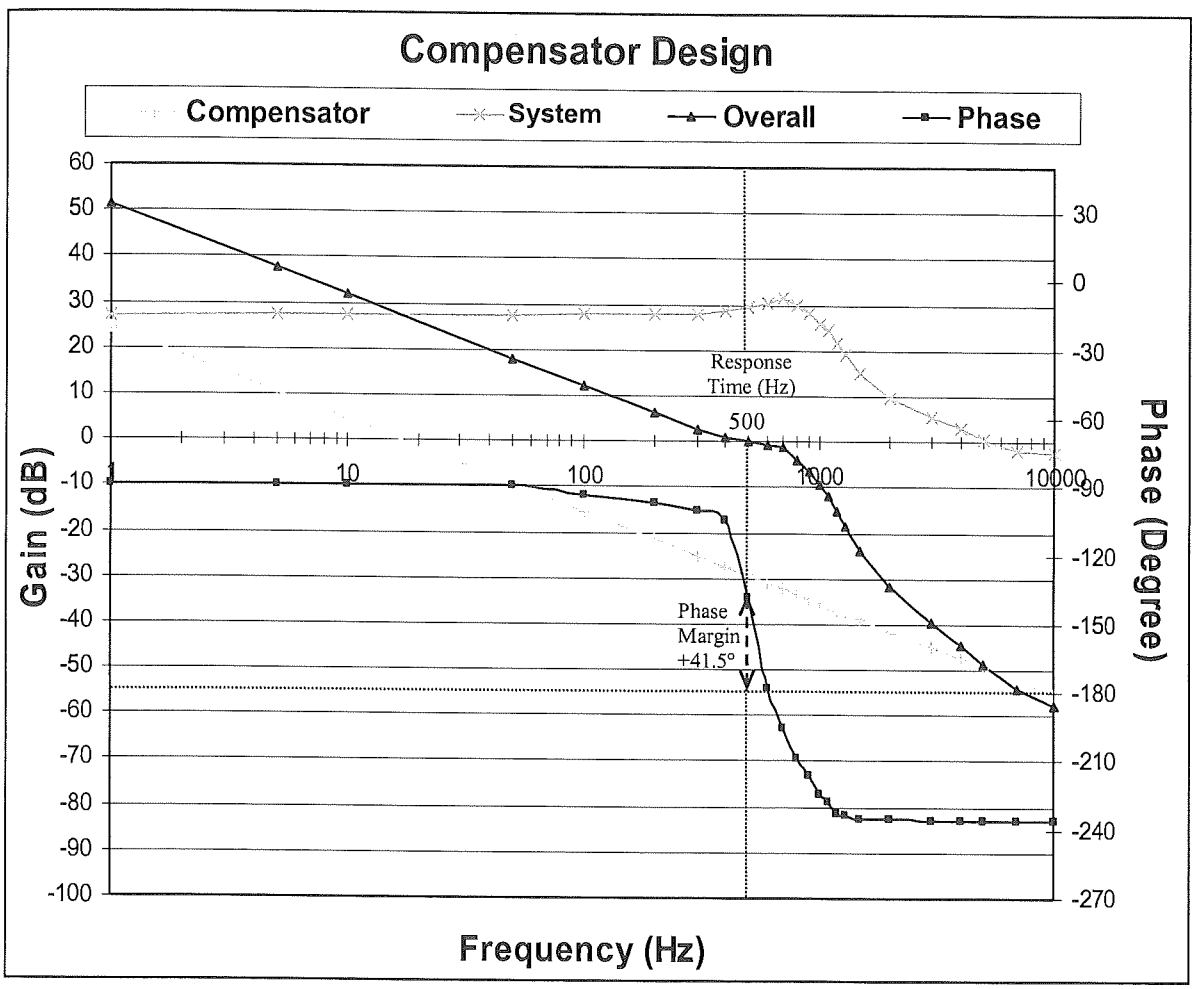


Figure 4.6 Bode-plot for the Cuk converter before and after the attachment of the Type one compensator.

As seen in Figure 4.6 there are four different coloured lines each representing the Type-one compensator (orange), the open loop system (green), the overall system response with compensation (blue) and the phase of the overall system (dark red).

A proper compensator is required to be designed to compensate the open loop system so that the overall system response has the following properties:

- An infinite DC gain. This allows the output to have zero steady-state error.
- A phase margin of around 45° but within the range from 30° to 60°.
- A slope of -20dB/decade at the crossover frequency.
- A crossover frequency that is lower than the resonant frequency. This simplifies the compensator design with a reasonable response time.

Referring to the system plant graph, it appears that the converter has a resonant frequency of 700Hz thus the overall system's response time of 500Hz is chosen. This is the compromise between a fast response and the complexity of the compensator design for the converter. A type one compensator is designed to compensate the open loop system so that the overall system is able to obtain a slope of -20dB/decade at the desired crossover frequency (see the overall system graph). From the phase of the overall system graph, it shows that the phase margin of the overall system is 41.5°, which means the overall system is stable.

4.5 Current Sensing and Pulse-by-Pulse Limiting Control

Current limiting control is used to prevent the converter from having over-current flowing through the power switching devices. There is a variety of current limiting strategies available depending on the application. Pulse-by pulse current limiting is selected since the UC3845 has it internally implemented. Figure 4.7 shows the current limiting operating region that is implemented in the Cuk converter.

As shown in Figure 4.7 when the current at the output is below 30A, the output voltage is regulated to 12V by the feedback compensation network as designed in Section 4.4. The pulse-by-pulse current limiting is enabled when the output current starts to exceed 30A. The output voltage will no longer be regulated at this point since the current feedback loop has overtaken the voltage feedback loop.

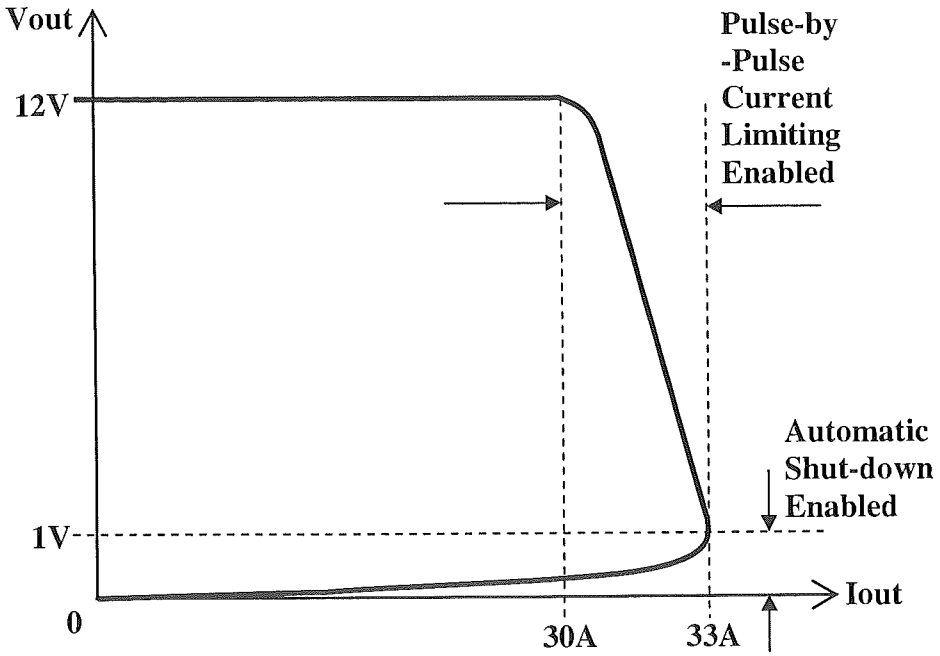


Figure 4.7 Planned current limiting strategy

The output voltage will drop significantly if the output load is increasing continuously. Since the output current is specified not to exceed 33A (±10% of 30A from Table 3.2), the automatic shut-down control is enabled to shut-down the converter if the output current is reaches 33A. This current limiting strategy provides an extra level of protection for the converter.

The current sense input of the current mode controller UC3842 is configured as shown in Figure 4.8 Current-to-voltage conversion of the MOSFET current is performed externally with ground-referenced current transformer. Note that the simplest way of sensing the current is to connect a number of low value resistors in parallel and measure the voltage drop. However, the power dissipation in these resistors is relatively large and it can be reduced significantly with the use of a current transformer. A current transformer with a turn's ratio of 1 to 50 is designed to step-down the expected peak MOSFET current of 40A.

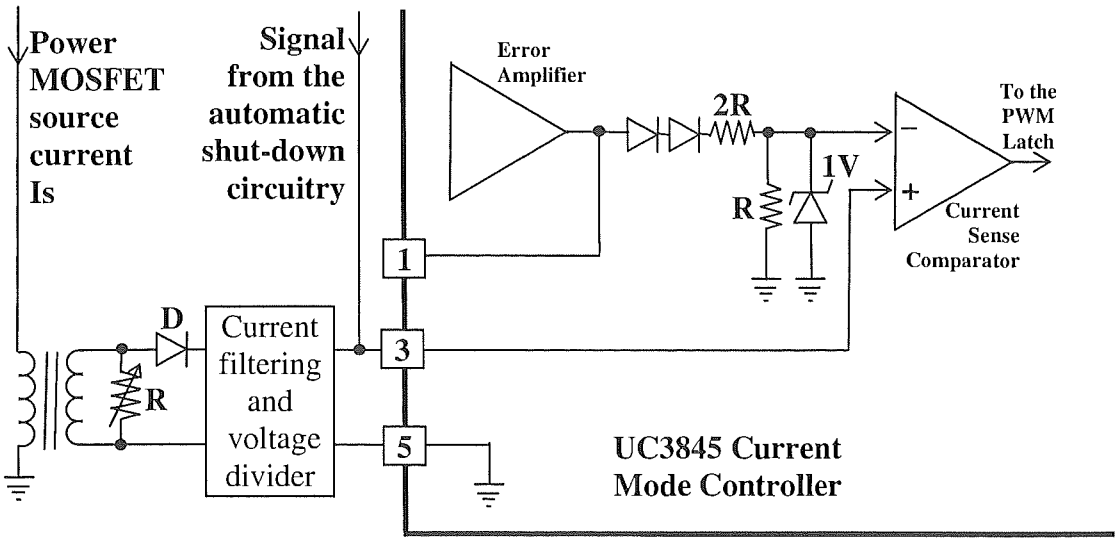


Figure 4.8 The UC3845 current sense input configuration

When sensing current in series with the power MOSFET as shown in Figure 4.9 the current will often have a large spike at its leading edge as shown in Figure 4.10. This is due to rectifier recovery and inter-winding capacitance in the current transformer. The transient can prematurely terminate the output pulse, in other words, this spike can turn off the MOSFET too early due to the false tripping. To solve this problem, a simple RC filter is inserted to suppress this spike. As an initial stage of designing the filter, the RC time constant should be approximately equal to the current spike duration. The current spike during is measured to be 94ns and therefore a resistor of 100Ω and a capacitor of 150pF are used as shown in Figure 4.9.

The inverting input to the UC3845 current-sense comparator is internally clamped to 1V (see Figure 4.8). The pulse-by-pulse current liming occurs if the feedback voltage at pin 3 (current sense input) reaches this threshold value. The voltage at pin 3 is designed to be proportional to the source current of the MOSFET switch so that it will only exceed 1V when 30A of the output current, or 40A of the source current, is reached. The 50Ω resistor is the terminating resistor which is used to convert the secondary current to voltage. The diode 1N4004 is used to DC-couple the AC voltage from the current transformer before filtering.

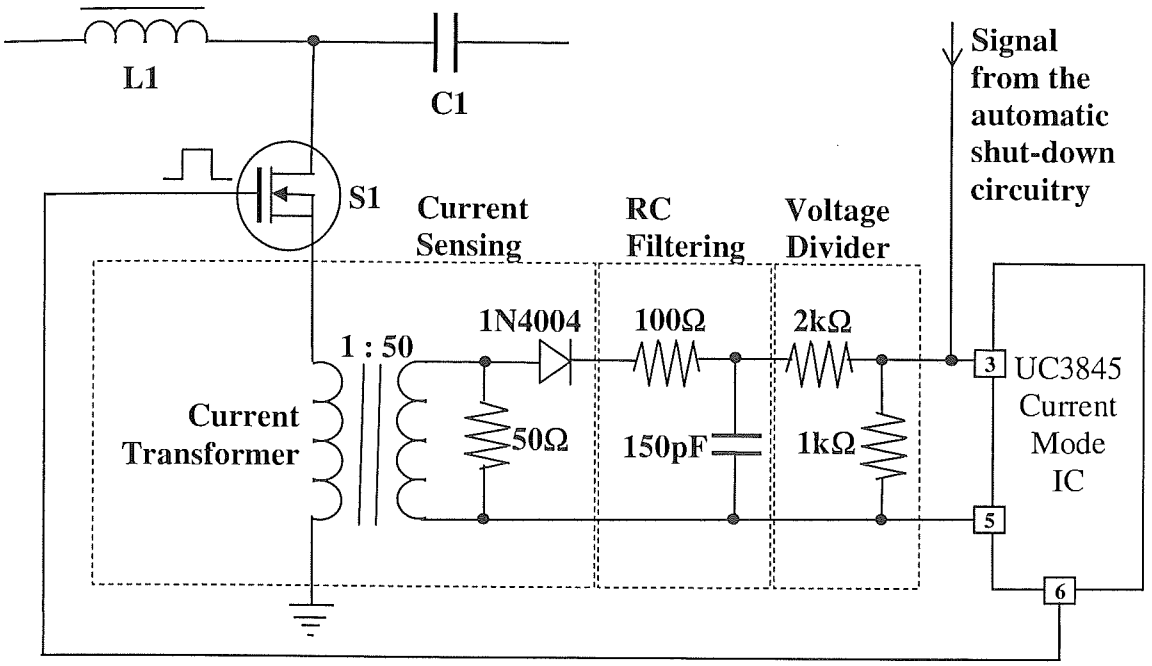


Figure 4.9 Transformer-coupled current sensing and filtering circuits.

As shown in Figure 4.10, it appears that the current spike is eliminated completely after the RC filtering. A small delay time of less than 10ns is found when comparing these two current waveforms. This delay is formed due to the phase shift of the RC filter and causing the output voltage unable to drop down to 0V when the current limiting is enabled. However, the automatic shut-down protection circuitry is designed to solve this problem and is presented in the next section.

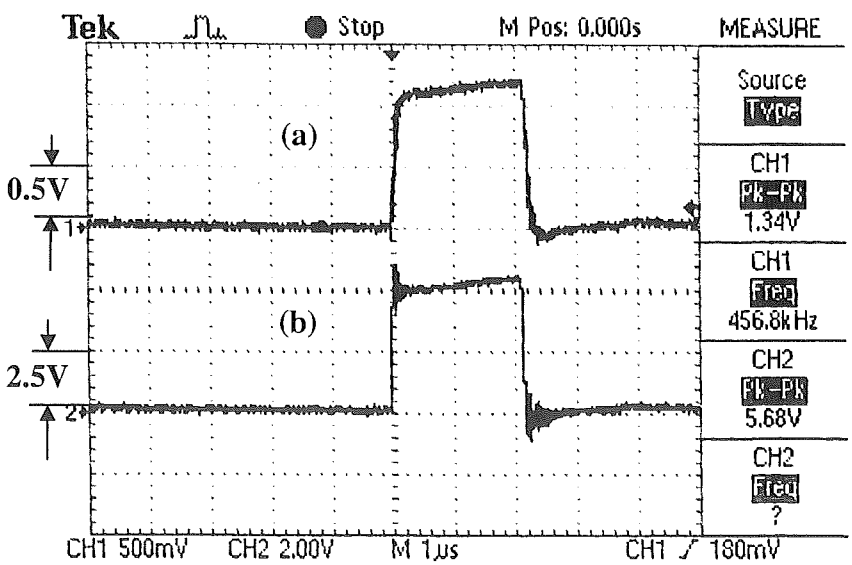


Figure 4.10 Source current sensed (a) after the RC filtered and voltage divided (0.5V/div, 1us/div), and (b) before the RC filtering (2.5V/div, 1us/div).

4.6 Automatic Shut-Down Protection

As shown in Figure 4.7, the automatic shut-down is designed to provide extra protection for the Cuk converter from the over-current at the output. It will replace the current limiting control when the output current is greater than 33A. Shut-down of the UC3845 IC is accomplished by raising the voltage of pin 3 (current sense input) above 1V. This method causes the output of the PWM comparator to be high (see Figure 4.3). The PWM latch is reset dominant so that the output will remain low until the next clock cycle after the shut-down condition at the pin 3 is removed.

Figure 4.11 shows the automatic shut-down circuitry of the converter. The shut-down is initialised when the LEM sensor sensed an excess output current and produces a signal to conduct the thyristor T1. This enables the PNP transistor S2 and generates a voltage of greater than 1V to the non-inverting input of the current sense comparator, which shuts down the PWM output signal. The shut-down condition can be released by pushing-open the manual switch S3.

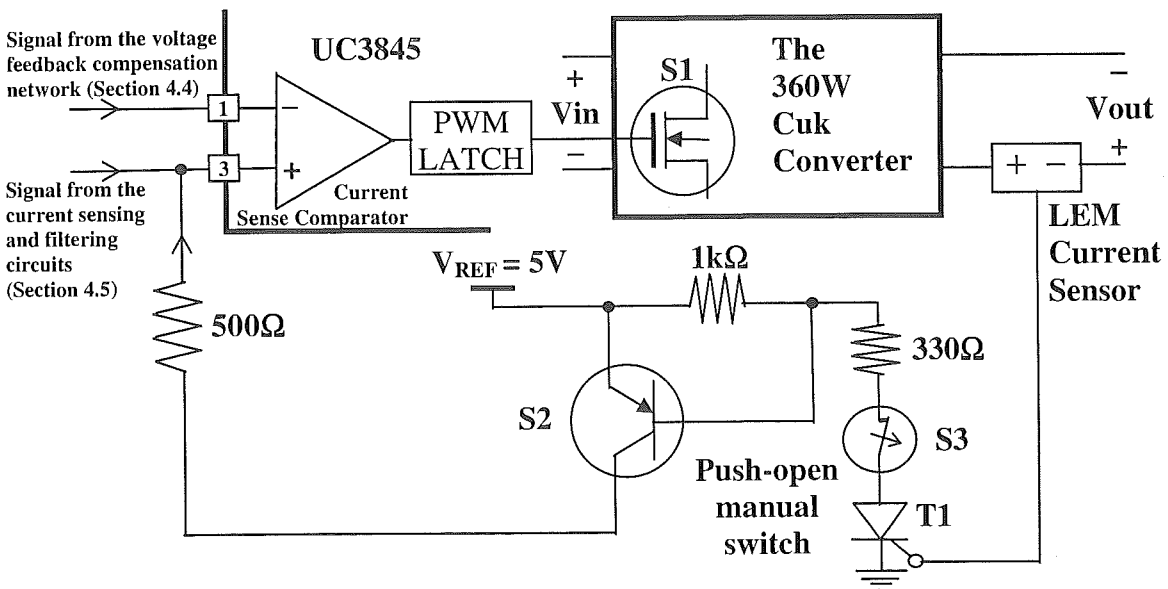


Figure 4.11 Automatic shut-down protection for the Cuk converter.

4.7 Construction

The PCB layout is of considerable importance in terms of reducing radiated EMI. It is also in the interest of good design to minimize stray inductances and so reduce voltage spikes within the circuit. Figure 4.12 shows a photograph of the completed 360W Cuk converter prototype. A double-sided PCB is used to provide a ground plane over the high current carrying tracks to minimize the loop area and therefore reduce the stray inductance. The PCB size is 21cm × 10cm with the power stage modules on the outer edge of the board makes it easier to mount the switching devices to the heatsink. The power stage layout maximizes the high current copper area on the PCB, as result, there is no bus bar in the design, and all the current is conducted on the PCB. Switching devices and passive components are labelled as shown in the figure.

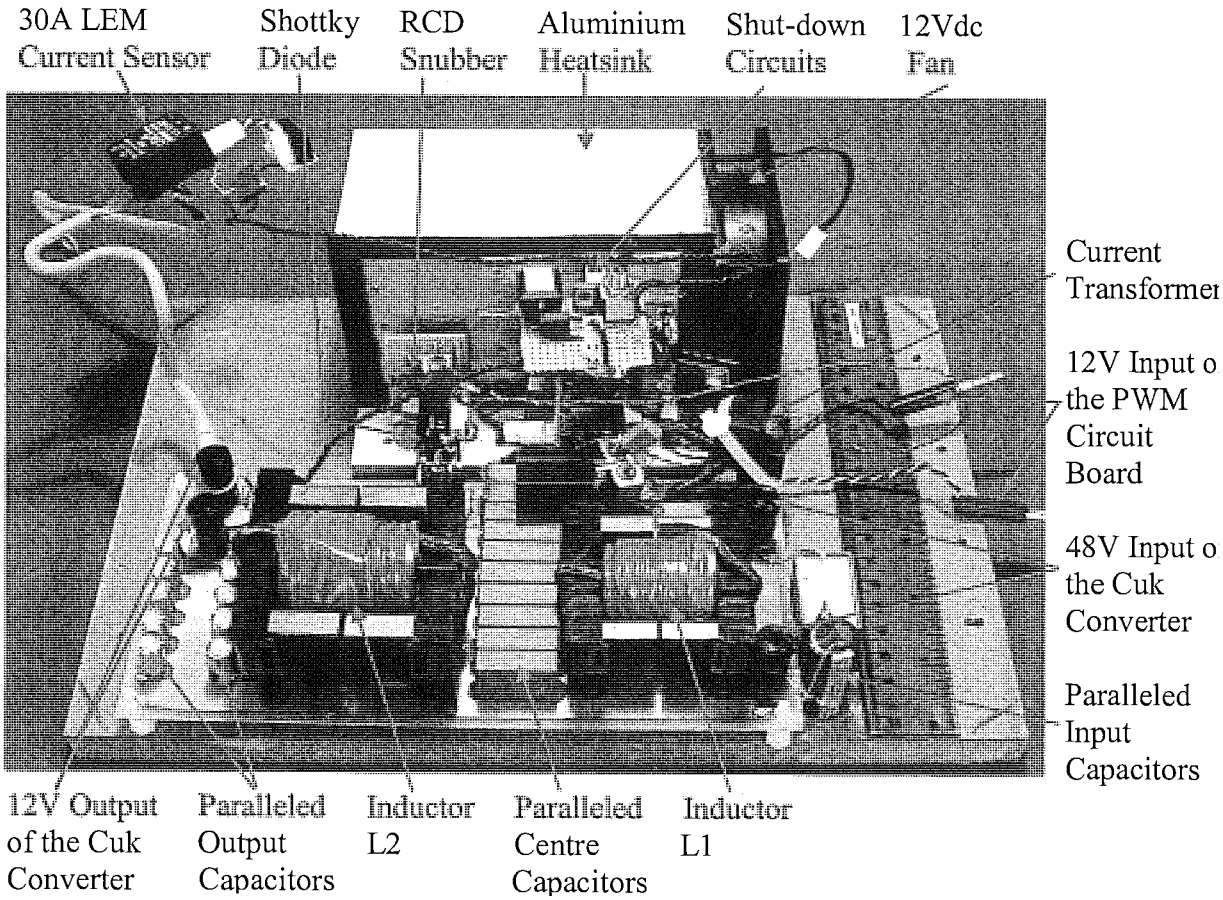


Figure 4.12 Photograph of the constructed 36W Cuk converter

4.8 Summary

The two PWM control mode strategies, the voltage mode and the current mode, are discussed. The current mode control strategy is selected for the application due to it simplifies the voltage feedback and the pulse-by-pulse current limiting loops. The current mode control circuitry consists of three feedback control loops: the voltage compensation, pulse-by-pulse current limiting, and the automatic shut-down protection. The voltage feedback compensation is designed with a response time of 500Hz. The current limiting control is set to activate once the output current exceeded 30A. If the output current is exceeding 33A, the converter will be turned off by the automatic shut-down circuitry to protect the converter from the over-current damage.

The prototype of the 360W Cuk converter is constructed completely. Double sided PCBs are made with ground planes to minimise the loop area and therefore reduce the stray inductance and improves efficiency. The performance of the prototype is evaluated by carrying out various experimental tests and the results are presented and discussed in Chapter 7.

The next stage of developing the Power Distribution Network is to design the CAN Control System to control the auxiliary loads in the MR2 and they are presented in the next chapter, Chapter 5.

5 CAN CONTROL SYSTEM DESIGN

Controller Area Network (CAN) is a well-designed serial communications bus for sending and receiving short real-time control messages. The CAN bus is designed to connect control systems over a small area. It can be operated in a noisy environment at data rates of up to 1 Mbps and has excellent error detection capabilities.

CAN was initially created by German automotive system supplier Robert Bosch in the mid-1980s for automotive applications, as a method for enabling robust serial communication. The goal was to make automobiles more reliable, safe and fuel-efficient, while decreasing wiring harness weight and complexity. Since its inception, the CAN protocol has gained widespread popularity in industrial automation and automotive/truck applications [13].

The CAN bus is a multi-master differential bus. Since the CAN protocol is message-based, the bus nodes do not have a specific address. Instead, the address information is contained in the identifier of the transmitted messages, indicating the message content and the priority of the message. The number of nodes may be changed dynamically without disturbing the communication of the other nodes.

The maximum CAN bus speed is 1Mbps, which can be achieved with a bus length of up to 40metres, when it is using a twisted wire pair, which is the most common bus medium used for CAN. Coaxial cable or optical fibre can also be used as the transmission medium. The relationship between data transfer rate and bus length is shown in Figure 5.1 [13]. For bus lengths longer than 40 metres the bus speed must be reduced, and at bus lengths above 1000 metres special drivers should be used.

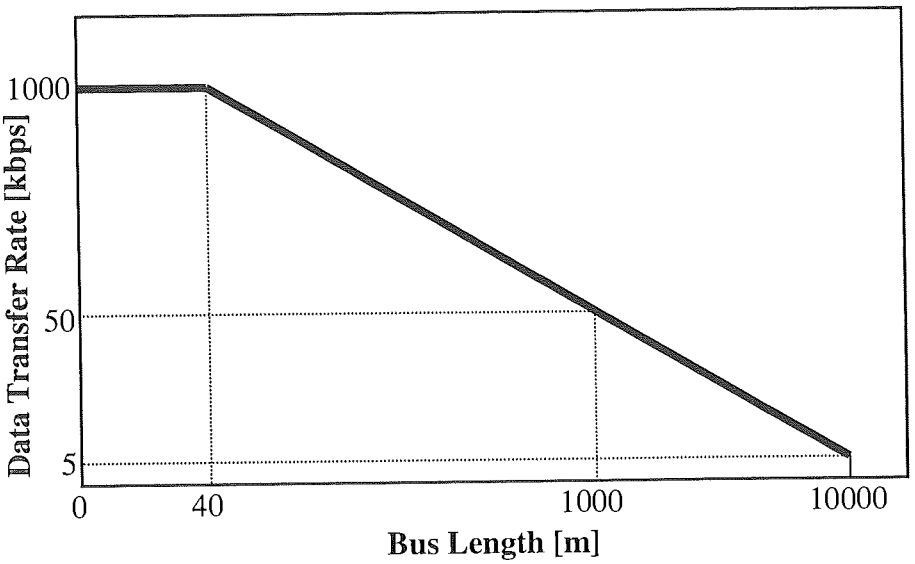


Figure 5.1 Relation between data transfer and bus length [13]

Since the CAN bus is going to be implemented in the MR2, the size of the vehicle is required to be investigated. The length and width of the MR2 are measured of 4.1 metres and 1.7 metres respectively. This means that approximately 10 metres of cable is required to run around the vehicle once. Therefore a total length of 20 metres of CAN bus cable is reserved for the MR2 application. According to the standard chart given in Figure 5.1, the maximum data transfer rate of 1Mbps can be achieved using a 20 metres length of cable. The design specification of data transfer delay on the CAN bus is not noticeable by the human eye. Therefore a data transfer rate of 125kbps on the CAN bus can be selected.

The following sections of this chapter describe the design of the Control System using CAN topology. The basics of the CAN protocol will be introduced, including its message format and properties. Then the reasons for selecting CAN to implement the Control System are made according to the strengths and advantages of this protocol. The typical implementation of CAN will be discussed. After that, the development of the CAN Control System will be presented and discussed in detail based on the specific design requirements.

5.1 The CAN Protocol

As mentioned earlier, the CAN protocol is a message-based, not address-based protocol. All nodes in the system receive every message transmitted on the bus and will acknowledge if the message is received successfully. It allows each node in the system to decide whether they want to ignore the message immediately or keep it to be processed.

CAN protocol is a CSMA/CD protocol. The CSMA stands for Carrier Sense Multiple Access. The Carrier Sense means that every node on the network must monitor the bus for a period of no activity before trying to send a message on the bus, and Multiple Access means once this period of no activity occurs, every node on the bus has an equal opportunity to transmit a message. The CD stands for Collision Detection, which means if two nodes on the network start transmitting at the same time, the nodes will detect the "collision" and take appropriate action. In CAN protocol, a non-destructive bitwise arbitration method is utilized. This means that collision of messages is avoided by bitwise arbitration without loss of time.

CAN nodes have the ability to determine fault conditions and transition to different modes based on the severity of problems being encountered. They also have the ability to detect short disturbances from permanent failures and modify their functionality accordingly. CAN nodes can transition from functioning like a normal node (being able to transmit and receive messages normally), to shutting down completely (bus-off) based on the severity of the errors detected. This feature is called Fault Confinement. No faulty CAN node or nodes will be able to take over all of the bandwidth on the network because faults will be confined to the faulty nodes which will be removed before bringing the network down. This is very powerful because Fault Confinement guarantees bandwidth for critical system information.

Unlike other bus systems, the CAN protocol does not use acknowledgement messages, but instead signals any errors immediately as they occur. The CAN bus provides sophisticated error detection and error handling mechanisms such as Cyclic Redundancy Check (CRC) and high immunity against electromagnetic interface. Temporary errors are recovered. Permanent errors are followed by automatic switch-off of defective nodes. Every

bus node in the system is informed about an error. This effectively guarantees that bandwidth will always be available for critical messages to be transmitted [14].

5.1.1 CAN Protocol Standards

CAN is internationally standardised by the International Organisation for Standardisation (ISO) as a template to follow for this layered approach. It is called the ISO Open Systems Interconnection (OSI) Network Layering Reference Model and is shown in Figure 5.2 [14].

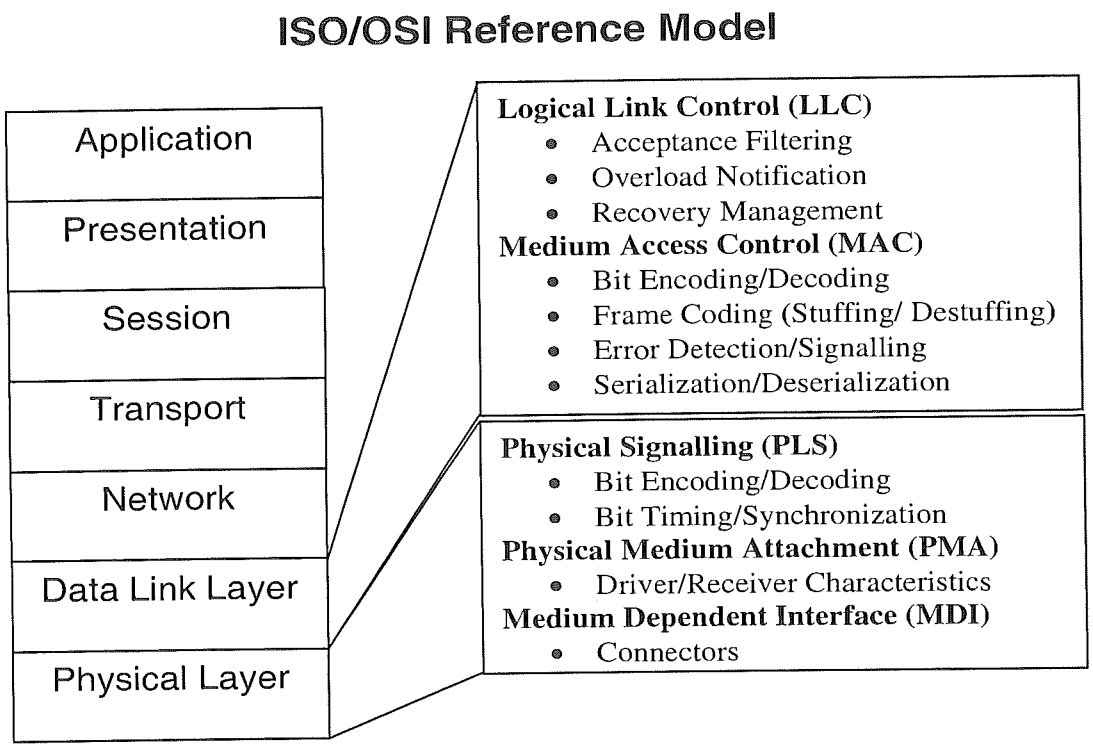


Figure 5.2 ISO/OSI Reference Model [14]

The CAN protocol uses the Data Link Layer and Physical Layer in the ISO/OSI model. There are also a number of higher-level protocols available for CAN. For the use of CAN two standards have been defined for the bus interface:

- CAN high speed according to ISO – 11898 for bit rates between 125kbps and 1Mbps.
- CAN low speed according to ISO – 11519 for bit rates up to 125kbps.

5.1.2 Message Frame Format

The CAN protocol supports two message frame formats, the only essential difference being in the length of the identifier. The so-called CAN standard frame, also known as CAN 2.0 A, supports a length of 11 bits for the identifier, whereas the so-called CAN extended frame, also known as CAN 2.0 B, supports a length of 29 bits for the identifier. Data and Remote Frames for these formats of CAN protocol are presented in Figure 5.3.

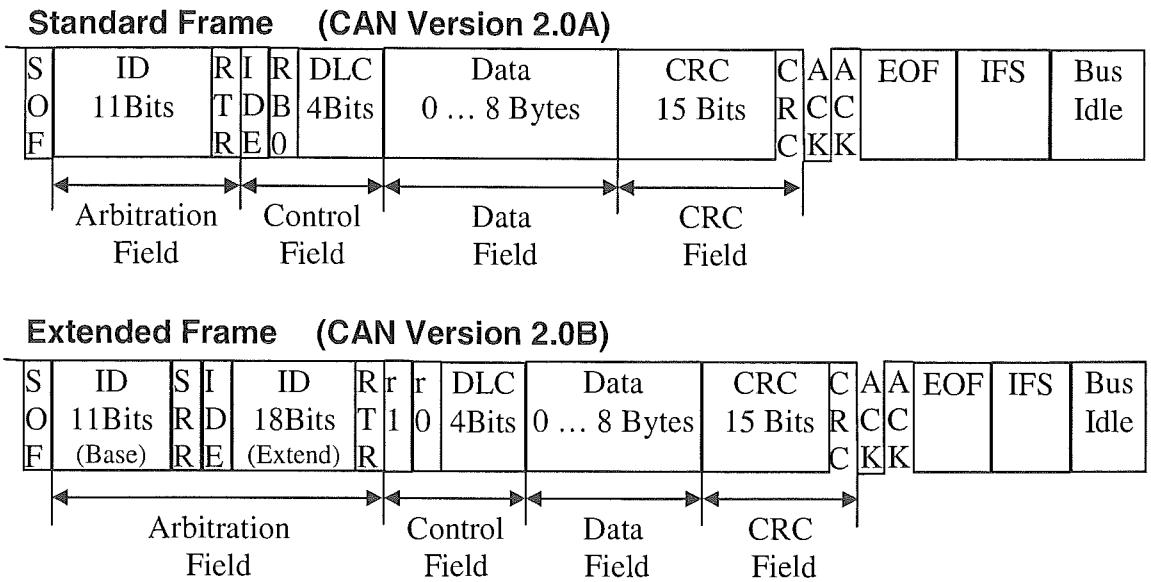


Figure 5.3 Standard and Extended frames format used in CAN

As shown in Figure 5.3, a message in the CAN standard frame begins with the start bit called Start Of Frame (SOF). This is followed by the "Arbitration Field" which consist of the IDentifier (ID) and the Remote Transmission Request (RTR) bit used to distinguish between the data frame and the data request frame. The "Control field" contains the 11 bit IDentifier Extension (IDE) bit to distinguish between the CAN standard frame and the CAN extended frame. The following bit, Reserved Bit Zero (RB0), is reserved and is defined to be a dominant bit by the CAN protocol. The Data Length Code (DLC) used to indicate the number of following data bytes in the "Data field". If the message is used as a remote frame, the DLC specifies the number of data bytes in the frame. The "Data field" that follows is able to hold up to 8 data bytes. The integrity of the frame is guaranteed by the following Cyclic Redundant Check (CRC) sum. The Acknowledge (ACK) comprises the ACK slot and the ACK delimiter. The bit in the ACK slot is sent as a recessive bit and is overwritten as a dominant bit by those receivers which have at this time received the data

correctly. Correct messages are acknowledged by the receivers regardless of the result of the acceptance test. The end of the message is indicated by End Of Frame (EOF). The Intermission Frame Space (IFS) is the minimum number of bits separating consecutive messages.

As mentioned before, a message in the CAN extended frame is likely to be the same as a message in CAN standard frame format. The only difference is the length of the identifier used. The identifier is made up of the existing 11-bit identifier (so-called base identifier) and an 18-bit extension (so-called extend identifier). The distinction between CAN standard frame format and CAN extended frame format is made by using the IDE bit, which is transmitted as dominant in case of a frame in CAN standard frame format, and transmitted as recessive in case of a frame in CAN extended frame format. As the two formats have to co-exist on one bus, it is laid down which message has higher priority on the bus in the case of bus access collision with different formats and the same identifier / base identifier: The message in CAN standard frame format always has priority over the message in extended format.

5.2 CAN Implementation Method

All CAN implementations have a common structure, consisting of a host micro-controller (MCU), CAN controller and CAN transceiver. There are differences, however, in the manner of integration of the above-mentioned components. The CAN nodes can either use stand-alone CAN controller to interface to a MCU, or an integrated CAN controller which already has a MCU and a CAN controller built-in as shown in Figure 5.4.

The reason for requiring a MCU alone with a CAN controller is that the CAN controller needs to be initialised by software programs. These programs are stored in the memory of MCU so the CAN controller can be accessed through the MCU interface, such as SPI. (This will be introduced in the following section). The CAN Controller handles all the transmitting and receiving of messages, which contain useful information for other nodes on the network via the CAN bus. The CAN transceiver is the interface between the CAN controller and the physical CAN bus. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller.

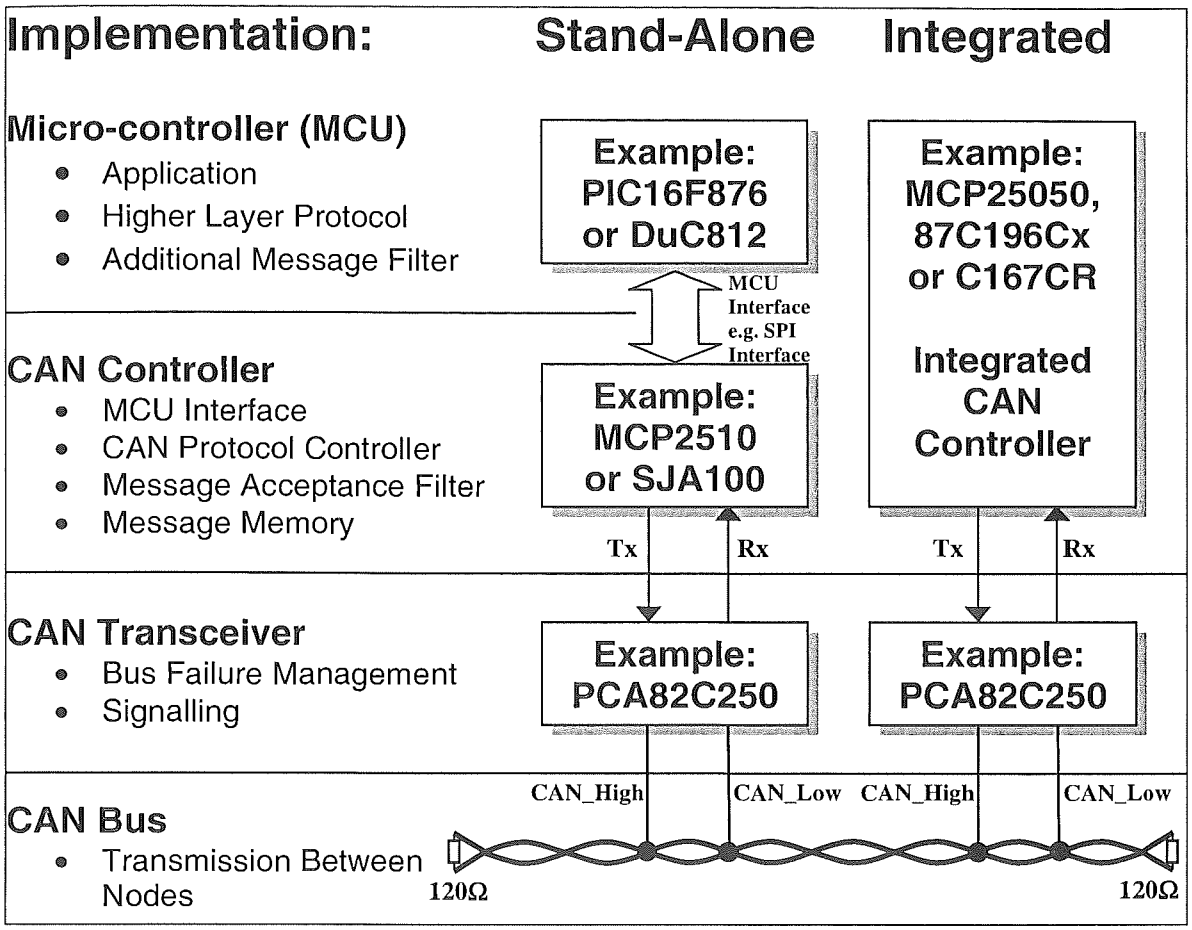


Figure 5.4 Possible structure of the CAN node

Circuits with stand-alone CAN chips are designed to interface to different MCUs, allowing the software developed for one system (for example in C language) to be reused in another system, even if the MCU is different. An architecture with integrated CAN controller causes a lower MCU load than stand-alone controllers (because of lower amount of time required to access to the CAN peripheral, following from fast internal address/data bus). Another important advantage of nodes using an integrated CAN controller over a system using stand-alone CAN with MCU is lower space requirements. On the other hand, software developed for the integrated CAN peripheral of one MCU may not apply to a second MCU with on-chip CAN, especially if the MCUs are supplied by different vendors. Thus, the benefits from a stand-alone CAN controller configuration is more favourable in the CAN Control System design because multiple designs will be urbanised in the future and the software and hardware development is reusable.

5.3 Serial Peripheral Interface (SPI)

As indicated in Figure 5.4, Serial Peripheral Interface (SPI) is the most common communication interface that is applied between the MCU and the CAN controller. SPI provides good support for the MCU to communicate with the CAN controllers and also some other peripheral devices such as Electrical Erasable Programmable Read Only Memory (EEPROM), Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC). It is a popular serial bus standard established by Motorola and supported in silicon products from various manufacturers. SPI interfaces are available on popular communication processors and micro-controllers. It is a synchronous serial data link that operates in full duplex, i.e. signal carrying data go in both directions simultaneously.

SPI is essentially a shift register that serially transmits data bits to other SPI's. During a data transfer, one SPI system acts as the ``master'' which controls the data flow, while the other system acts as the ``slave'' which has data shifted into and out of it by the master. Different MCU's can take turn being masters, and one master may simultaneously shift data into multiple slaves. However, only one slave may drive its output to write data back to the master at any given time.

SPI specifies four signals: Signal Clock (SCLK); Master data Output, Slave data Input (SI); Master data Input, Slave data Output (SO); and Slave Select (CS). Figure 5.5 shows these signals in a single slave configuration. SCLK is generated by the master and input to all slaves. SI carries data from master to slave. SO carries data from slave back to master. A slave device is selected when the master asserts its CS signal. In this project, single master, single slave SPI implementation is applied.

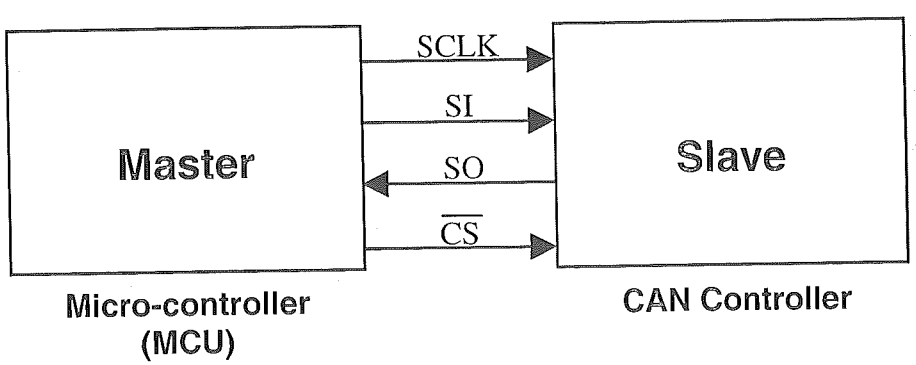


Figure 5.5 Single master, single slave SPI implementation

5.4 CAN Control System Design

5.4.1 Requirements

In order to control the auxiliary loads using the power provided by the Auxiliary Power System as discussed in the Chapter 2, an efficient and fast communication link (such as CAN) is desired. Following is a list of the design requirements.
A prototype must be able to:

- Switch auxiliary loads on and off as desired.
- Sufficient control outlets for all the auxiliaries in the vehicle.
- Ability to detect load faults during operation.
- Provide robust and accurate control in a noisy environment (as in the vehicle) with up to 40 metres of wiring distance.

5.4.2 Implementation of the CAN Control System

As mentioned in the beginning, because of its properties, the CAN bus can find practical application in the control area of the Power Distribution Network for the electric vehicle, that is, the CAN Control System. Figure 5.6 shows the practical implementation of the CAN Control System. The control system consists of five CAN nodes, based on the Microchip PIC micro-controllers, PIC16F876 and PIC16F877, and the stand-alone CAN controller, MCP2510. Table 5.1 lists the key features of these devices.

The MCP2510 is chosen because it features an industry standard SPI serial interface, enabling an easy connection to virtually any micro-controller that has a SPI interface built-in. MCP2510 is also the easiest to use, most cost-effective, stand-alone CAN controller on the market today. The PIC16F876 and PIC16F877 MCUs are selected due to the support of SPI interface ports and FLASH / EEPORM fast memory access.

Table 5.1 Key Features of the MCU and CAN controller used.

Device / Part Number / Manufacturer	Key Features
Micro-Controller (MCU) PIC16F876 PIC16F877 Microchip.com	• Low power, high speed CMOS FLASH / EEPROM technology • Synchronous Serial Port (SSP) with SPI and I^2C supported • Single 5V In-Circuit Serial Programming (ICSP) via two pins • Operating speed of DC 20MHz clock input • Wide operating voltage range from 2.0V to 5.5V.
Stand-Alone CAN Controller MCP2510 Microchip.com	• Lowest pin count stand-alone CAN controller • Simple SPI Interface • Full CAN 2.0B active implementation • Minimise MCU overhead requirements - Multiple transmit and receive buffers - Masks and filters limit messages to be processed • Buffered clock output • Hardware and software message triggers

As shown in Figure 5.6, there are two types of CAN nodes (so-called CAN Control Node and CAN Load Node) along the CAN bus. Between every CAN Load Node and the 12V auxiliary loads, a logic circuitry (so-called Load Fault-Detector) is designed to switch loads and detect the auxiliary loads during the operation.

The configurations of the CAN Control Node and CAN Load Node are identical; the only differences are in the choice of MCU chipset and the circuitry that they are designed to interface to. That is, the CAN Control Node uses a 40-pin PIC16F877 MCU and sense signals from the switch controller. While the CAN Load Node use a 28-pin PIC16F876 and is designed to interface to the Load Fault-Detector. Both of the nodes are be able to transmit and receive messages via the CAN bus.

The PIC micro-controller with integrated FLASH / EEPROM stores program in memory and generating messages, depending on commands incoming from either the switch controller or other side of the CAN bus.

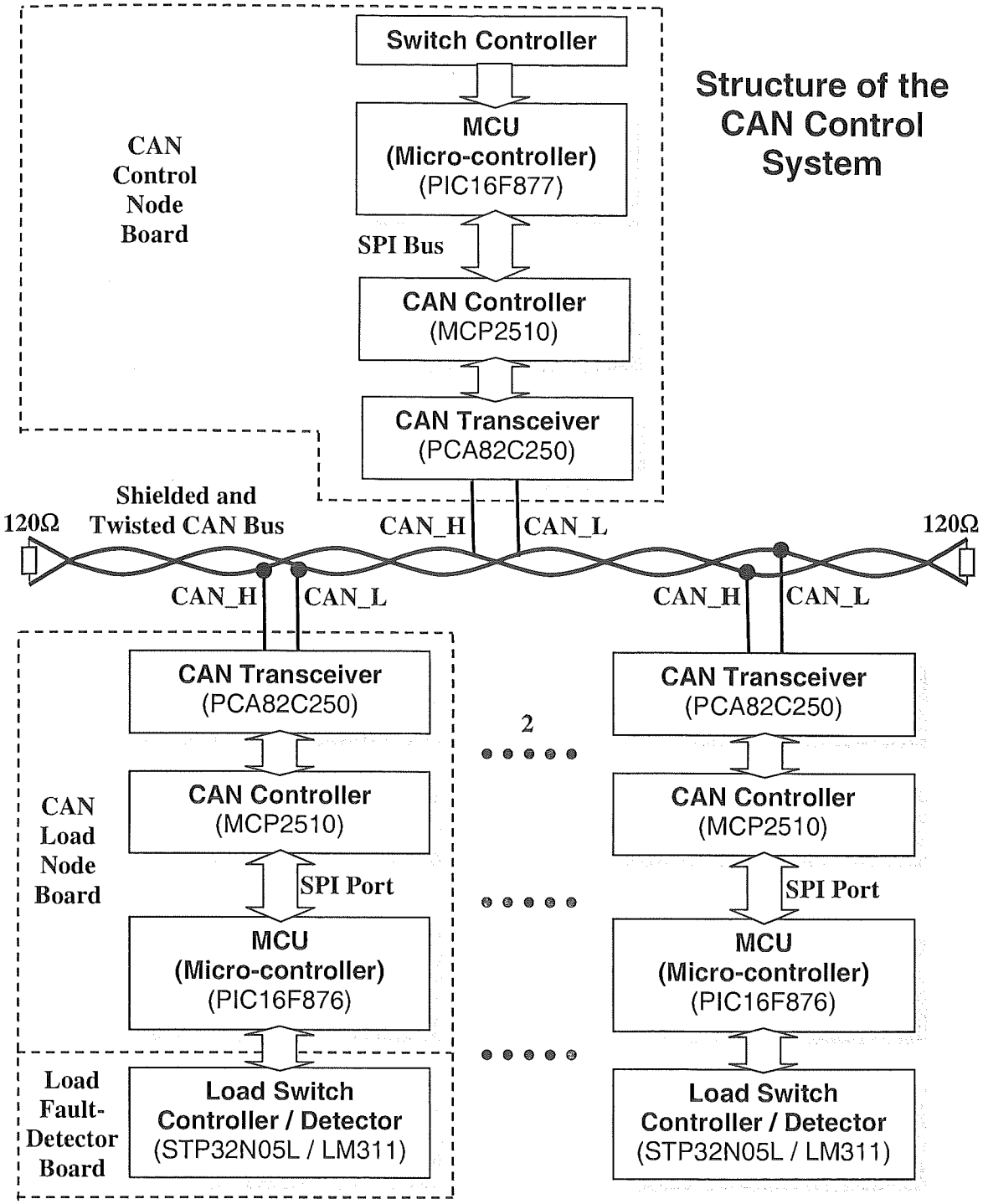


Figure 5.6 CAN Control System in the EV

The CAN protocol controller chip in each node – MCP2510, this is the heart of the CAN interface. It is responsible for handling all the messages transferred and received via the CAN bus. One of the most significant features of the MCP2510 CAN controller is the acceptance filter capability. Figure 5.7 shows a block diagram of the receive buffer built into the MCP2510 [15].

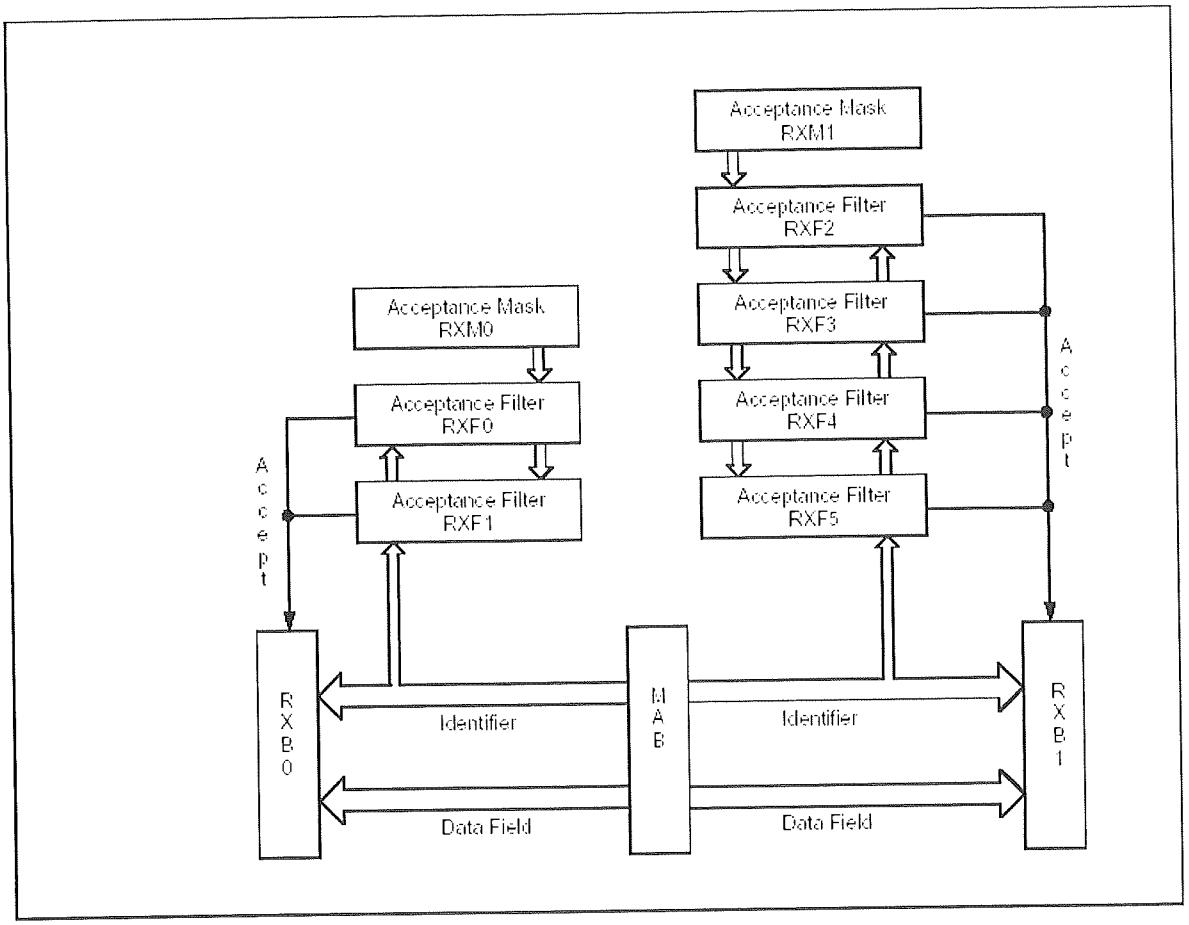


Figure 5.7 Receive buffer block diagram [15]

Message filtering is used for the acceptance criteria of each node. Each node had five acceptance filters (RXF1~RXF5) and two acceptance masks (RXM0 and RXM1) as shown in Figure 5.7. These filters are associated with two receiver buffers (RXB0 and RXB1). The buffer with fewer filters had a higher priority than the other. Every message is loaded into the Message Assembly Buffer (MAB) and is only transferred to the receive buffer if the identifier matched the filter, which means that multiple nodes can accept the same message. However, the hardware acceptance filters are supported by software in the micro-controller; passes only needed messages, throwing away commands destined to other nodes.

The CAN controller is coupled with the physical bus by the CAN transceiver, PCA82C250, which provides differential transmit capability to the bus and differential receive capability to the CAN controller.

The Load Fault-Detector consists of four sets of logic level switches (STP32N05L) and OP-AMP Comparators (LM311). It is designed not only to switch on and off the auxiliary loads, but also to monitor the status of loads during the operation. This is done by sensing the source current of the MOSFET switches. An interrupt signal will be sent to the MCU via CAN bus if no current is detected while the switch is turned on.

The ISO-11898 assumes the network wiring technology to be close to a single line structure in order to minimise reflection effects on the bus line. The bus lines have to be terminated by resistors at both ends. In the implementation of CAN Control System, between 15 to 30 metres of shielded twisted pair ended with 120 Ohm resistors will be used. All CAN nodes along the CAN bus send and receive commands via CAN bus with the bit-rate of 125kbps. Each command, as a Standard Format of CAN Message (see Figure 5.3), has got an address of the CAN node that the message is destined to. The CAN nodes execute only these commands, whose address correspond to ID address of the CAN node. All the further details in this section will be carried out in the following chapter (Chapter Six).

5.5 Summary

The CAN protocol is optimised for systems that need to transmit and receive relatively small amount of information reliably to any or all other nodes on the network. The use of CSMA/CD allows every node to have an equal chance to gain access to the bus. Since the protocol is message-based, all nodes on the bus receive every message and acknowledge every message, regardless of whether the node required the data or not. Fast robust message transmission with fault confinement is the big plus for CAN because of faulty nodes will automatically drop off the bus, which does not allow any faulty node to bring down the network. This effectively guaranteed that bandwidth will always be available for critical messages to be transmitted. With all these benefits built into the CAN protocol and its momentum in the automotive world, the CAN topology is selected and implemented into the Auxiliary Distributed System for the MR2. The CAN protocol supports two message frame formats, the standard frame format (CAN 2.0A) and the extended frame format (CAN 2.0B). Each frame format consists of various fields and bits and descriptions are

provided. The CAN 2.0A was selected as the CAN protocol format in the MR2 application due to the message in CAN standard format always has priority over the extended format.

CAN hardware implementation methods were studied and the stand-alone CAN controller configuration was chosen. This is because with this configuration, multiple designs could be developed in the future and the software and hardware development is reusable. The CAN Control System was designed within the design specifications and consists of five CAN nodes, based on the PIC16F87X MCU and MCP2510 CAN controller. All the details on the construction of the CAN Control System are presented in the following chapter, Chapter 6.

6 CAN CONTROL SYSTEM CONSTRUCTION

This chapter describes the development and implementation of the CAN Control System in the Power Distribution Network for the MR2 electric car. The CAN Control System hardware is broken down into three components as shown in Figure 5.5, and these are:

- CAN Control Node,
- CAN Load Node, and
- Load Fault-Detector.

The CAN Control Node has the capability of monitoring eight analogue inputs from the switch controller and automatically generating messages based on their values. These messages will then be transmitted to the CAN Load Nodes via the CAN bus. The CAN Control Node is also capable of controlling eight analogue outputs, responding to message requests from the CAN Load Nodes via the CAN bus and generating time-based messages to turn on and off the Load Fault LEDs.

Each CAN Load Node is identical to each other in terms of PCB layouts and they have similar circuitry compared to the CAN Control Node. The major difference is the use of the PIC MCU as mentioned in Section 5.4.2. The CAN Load Nodes have the ability of receiving signals from the Load Fault-Detector Boards and automatically generating message based on the received signals. These messages are then be transmitted back to the CAN Control Node and light the load fault LEDs via the CAN bus. The CAN Load Nodes are also capable of controlling up to sixteen analogue outputs, responding to message request from the CAN Control Node via the CAN bus to turn on and off the MOSFETs in the Load-Fault Detectors.

The Load-Fault Detectors are implemented to monitor the status of the auxiliary loads using a current sensing technique. Each board has the ability for controlling up to four individual auxiliary loads, responding to message request from the CAN Load Nodes. A +5V signal is generated and transmitted to the CAN Load Node if zero MOSFET current is detected when the load is turned on. Therefore, this signal represents a load fault condition.

The designed CAN Control System supports a maximum CAN bus speed of 125kbits, with both standard and extended frames. The system is presented using standard frames. Some changes in the programme code would be required to implement extended frames in the future. The following sections of the chapter focus on the development and construction of the CAN Control System hardware. This includes the functionality description of each individual CAN node followed by the construction of these nodes. Then the software of the CAN Control System is presented along with a flow chart diagram. Finally, a summary is included to summarise the development of the CAN Control System hardware and software presented in this chapter.

6.1 CAN Control Node

This section describes the CAN Control Node and how the CAN functions in this node. The high level design of the CAN Control Node board is shown in Figure 6.1. The concept is to enable the MCP2510 CAN controller and the PIC16F877 micro-controller (MCU) to efficiently communicate among each other utilising the SPI. The MCP2510 handles lower level protocols.

The PIC MCU stores the program in memory and reads the settings of the 8 input switches. Any of the first 4 switches tell the PIC MCU to generate a message byte using the CAN standard frame with an ID of "0" (IDentifier, see Section 5.1.2). This identifies the destination of the message, in other words, the message generated by these 4 switches will only be accepted by the CAN Load Node 1. Consequently, CAN Load Node 2 only read a message with an ID of "1", which is set by any of the last 4 input switches. The Switch LEDs are mounted to indicate that the switches are turned on and the Load-Fault LEDs are used to indicate there is no load current that have occurred at the output loads.

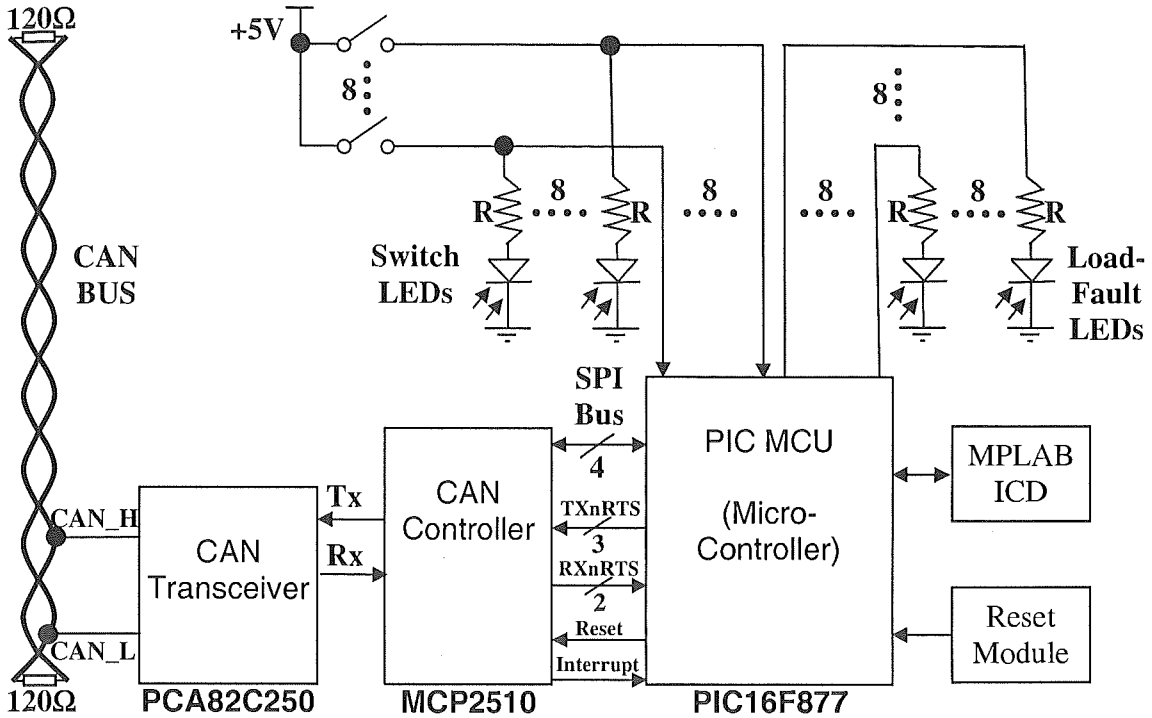


Figure 6.1 Detailed block diagram of the CAN Controller Node

There are two options for the PIC MCU and the MCP2510 to communicate each other. The first option is to use the 4 line SPI bus to handle all the message transmission, reception and error detection. The second option is to allow several links to be in charge of these tasks individually, such as using the TXnRTS (Transmit buffer Request To Send) for transmitting messages, the RXnRTS (Receive buffer Request To Send) for receiving messages, the Reset for resetting MCP2510 and the interrupt to indicate message errors. Due to the SPI's advantages mentioned in Section 5.3, the second option is not used in the MR2 application. However, the extra wiring connections are made on the PCB for the future use of the second option.

As seen in Figure 6.1, there are two modules connected to the PIC MCU: The MPLAB-ICD and the Reset module. The Microchip's MPLAB-ICD (In-Circuit Debugger) is the tool that is used to enhance the code development and software debugging process in the node boards. The ICD uses a PIC16F877 device and operates in "real time". This tool allows the application program and circuit to be evaluated and enhanced in real time. The ICD interface also allows the PIC MCU to be programmed after the board has been manufactured. This allows software changes or updates to be programmed into device

through a RJ45 (telephone jack) connection. The ICD uses the RB6 and RB7pins of the PIC MCU for this. For that reason, these pins are not used for any other purpose in the system.

Once the message generated by the PIC MCU is stored in the MCP2510's control register. The message will then be transmitted onto the CAN bus via the PCA82C250 CAN transceiver once the "Request to send" instruction is received from the MCP2510. There is no need for the software to operate the CAN transceiver. The transceiver has a pair of connection at both input and output to transmit/receive differential CAN messages via the CAN bus as shown in Figure 6.1.

The detailed circuit diagram of the CAN Control Node board is included in Appendix C1. The CAN Control Node board contain both digital and analogue circuitry. This required careful layout of the board to ensure that no digital noise is inadvertently coupled into the analogue circuitry.

Figure 6.2 show the photograph of the completed CAN Control Node board. A doubled sided PCB is used with a ground plane on top and +5V source plane on bottom with tracks are on both side of the PCB to minimize the loop area and therefore reduce the stray inductance. A DB9 and a RJ45 jack are mounted on the board for connecting the CAN bus and the MPLAB-ICD module respectively. There are numbers of test pins placed beside the PIC MCU and the MCP2510 for the purpose of sensing signals during testing. A push-to-close reset button is mounted on the PCB for resetting the PIC MCU. Different colours of LEDs are placed around the board to display the status of the switches and the output loads.

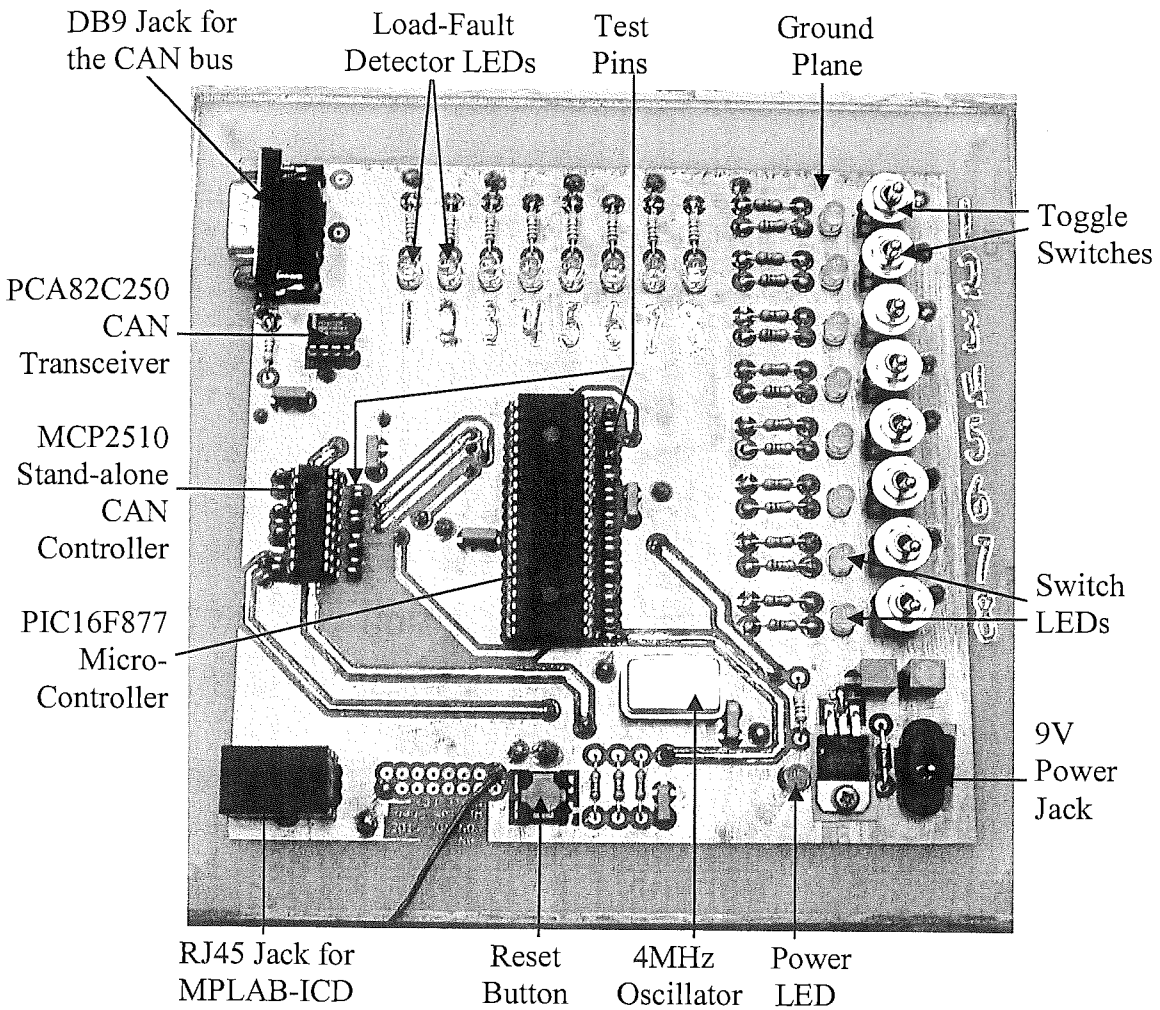


Figure 6.2 The CAN Control Node Board.

6.2 CAN Load Node

As mentioned in Section 5.4.2, the configuration of the CAN Load Node is identical to the CAN Control Node. The only differences are in the choice of MCU chipset and the circuitry that they are designed to interface to. The CAN Load Node use a 28-pin PIC16F876 MCU because there are only 4 inputs and outputs required on each load node. The CAN Load Node is designed to interface to the Load Fault-Detector board and the CAN messages will be generated according to the signal received from the board.

Figure 6.3 shows the circuitry of the CAN Load Node in block diagram. The software written for the CAN Load Node is identical to the CAN Control Node. This makes

the process of transmitting and receiving CAN messages the same. The detailed circuit diagram of the CAN Load Node is included in Appendix C2.

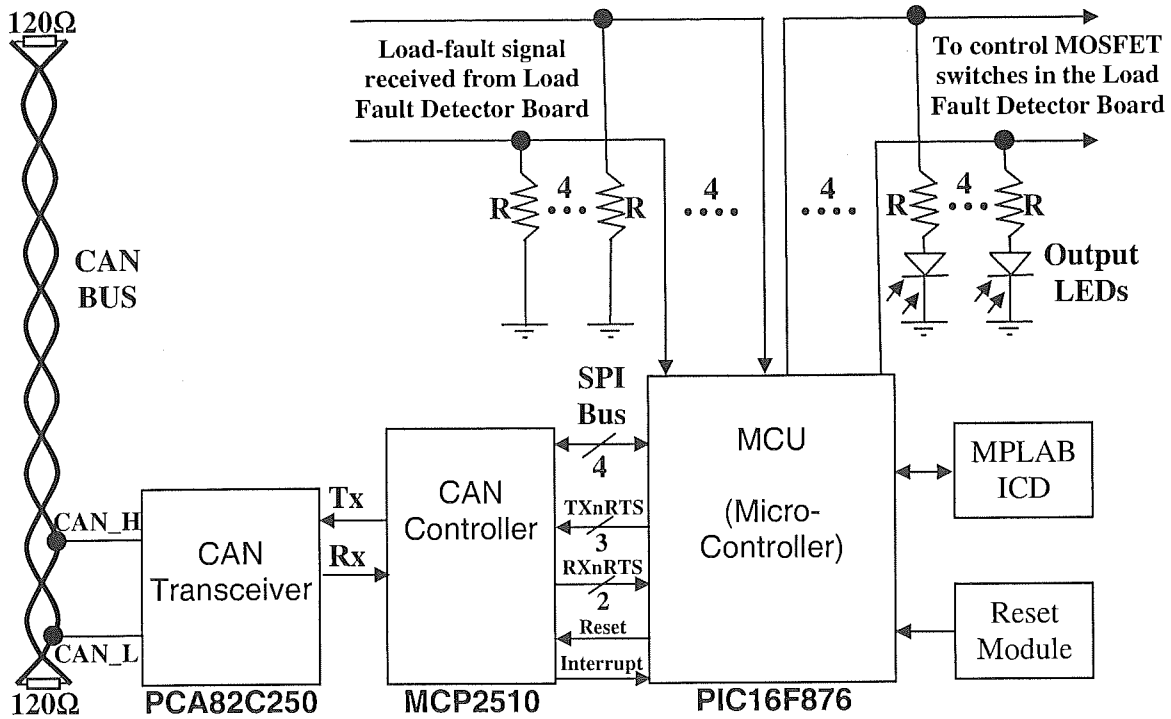


Figure 6.3 Detailed block diagram of the CAN Load Node

Two CAN Load Node boards have been constructed and Figure 6.4 shows a photograph of one of the completed CAN Load Node boards. From the figure, it can be seen that the board is similar to the CAN Control Node board (see Figure 6.2). The only difference is that instead of placing toggle switches, numbers of headers are mounted on the board for the purpose of transmitting and receiving signals from the Load Fault-Detector board.

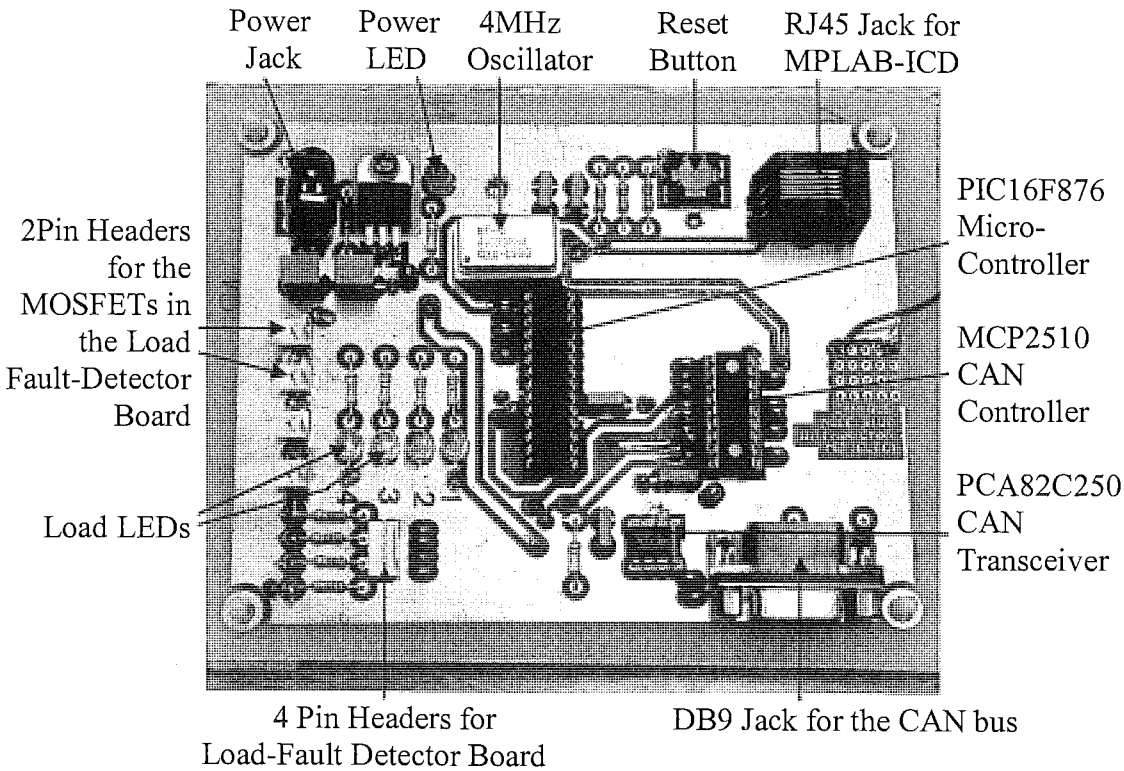


Figure 6.4 One of the completed CAN Load Node Boards

6.3 Load Fault-Detector

The Load Fault-Detector is developed to turn on and off up to four 12V auxiliary loads. An additional feature of the board is to generate a +5V signal when a fault is detected within one of the auxiliary loads. This signal is then sent back to the CAN Load Node. Figure 6.5 shows the designed circuitry of the board in block diagram form. Four of logic-level MOSFETs, STP32N05L, are used to turn on and off the loads. These MOSFET switches are controlled by the +5V signals received from the CAN Load Node board. Four toggle switches are used to simulate open-circuit load faults. To detect the load-fault, a set of ten 1Ω resistors are connected in parallel to sense a MOSFET's current and convert it to a voltage signal. This voltage signal is then feed in to the inverted input of an OP-AMP comparator, LM311, as shown in Figure 6.5. If one of the loads is faulty, the circuitry will become open-circuited which provides a 0V sensed voltage signal. The comparator will then produce a +5V analogue signal and transmits it to the CAN Load Node board as the sensed voltage signal is lower than the reference voltage of the comparator.

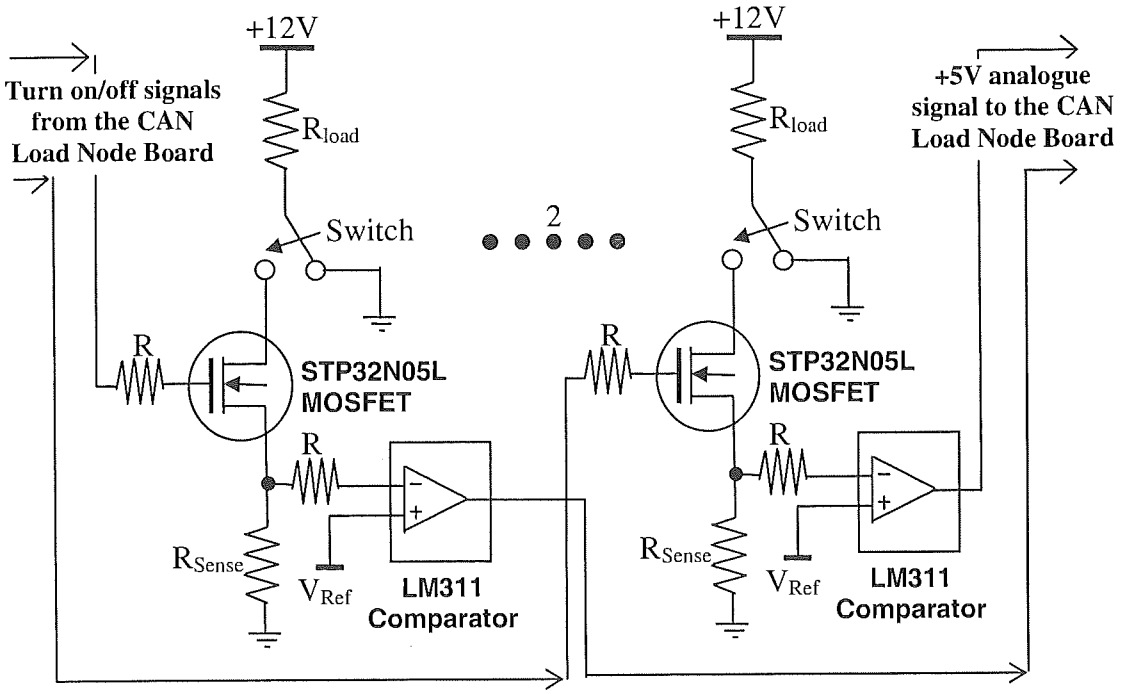


Figure 6.5 Detailed block diagram of the Load Fault-Detector

The detailed circuit diagram of the Load Fault-Detector board is included in Appendix C3. Two Load Fault-Detector boards are constructed and Figure 6.6 shows the photograph of the one of the completed Load Fault-Detector boards. The PCB is double side constructed with ground plane on top and +12V source plane on bottom with tracks are on both side of the PCB to minimize the loop area and therefore reduce the stray inductance. From the figure, it can be seen that there are three 10W, 33Ω resistors connected in parallel between each set of current sensing resistors and a toggle switch. These resistors are mounted on the board to provide extra resistive loading during the 360W converter tests as will be presented in Chapter 7. Two pin headers are placed close to the MOSFETs to reduce the stray inductance. Four sets of power outlets are also mounted on the board for easy connection of the auxiliary loads in the MR2.

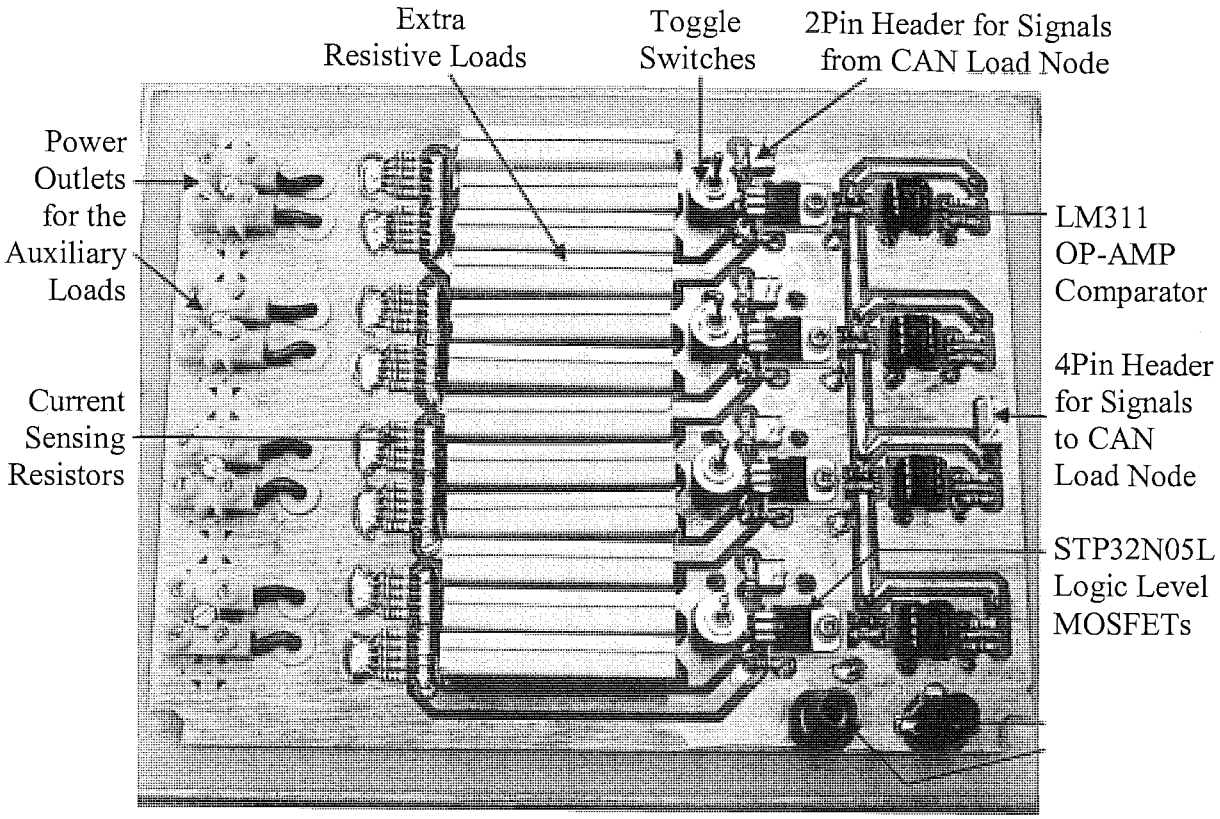


Figure 6.6 One of the completed Load Fault-Detector Boards

6.4 System Software

The software for the CAN Control Node and the CAN Load Node are written in C and installed into the PIC MCUs using the Microchip's MPLAB. Full listings of the software are included in Appendix C4 and Appendix C5. As mentioned in Section 6.2, the software for the CAN Control Node and the CAN Load Node are identical to each other as their processes of transmitting and receiving CAN messages are the same. The compiled code occupies just less than 1k byte of program memory.

Figure 6.7 shows the main program flow of the software. On power-up the registers within the MCU and the MCP2510 are initialised. The first part of this process involves setting up all the I/O pins of the MCU as either inputs or outputs depending on their function, configuring the SPI bus for serial communication between the MCP2510 running at 4MHz. This is achieved by writing the appropriate information to the registers associated with the port.

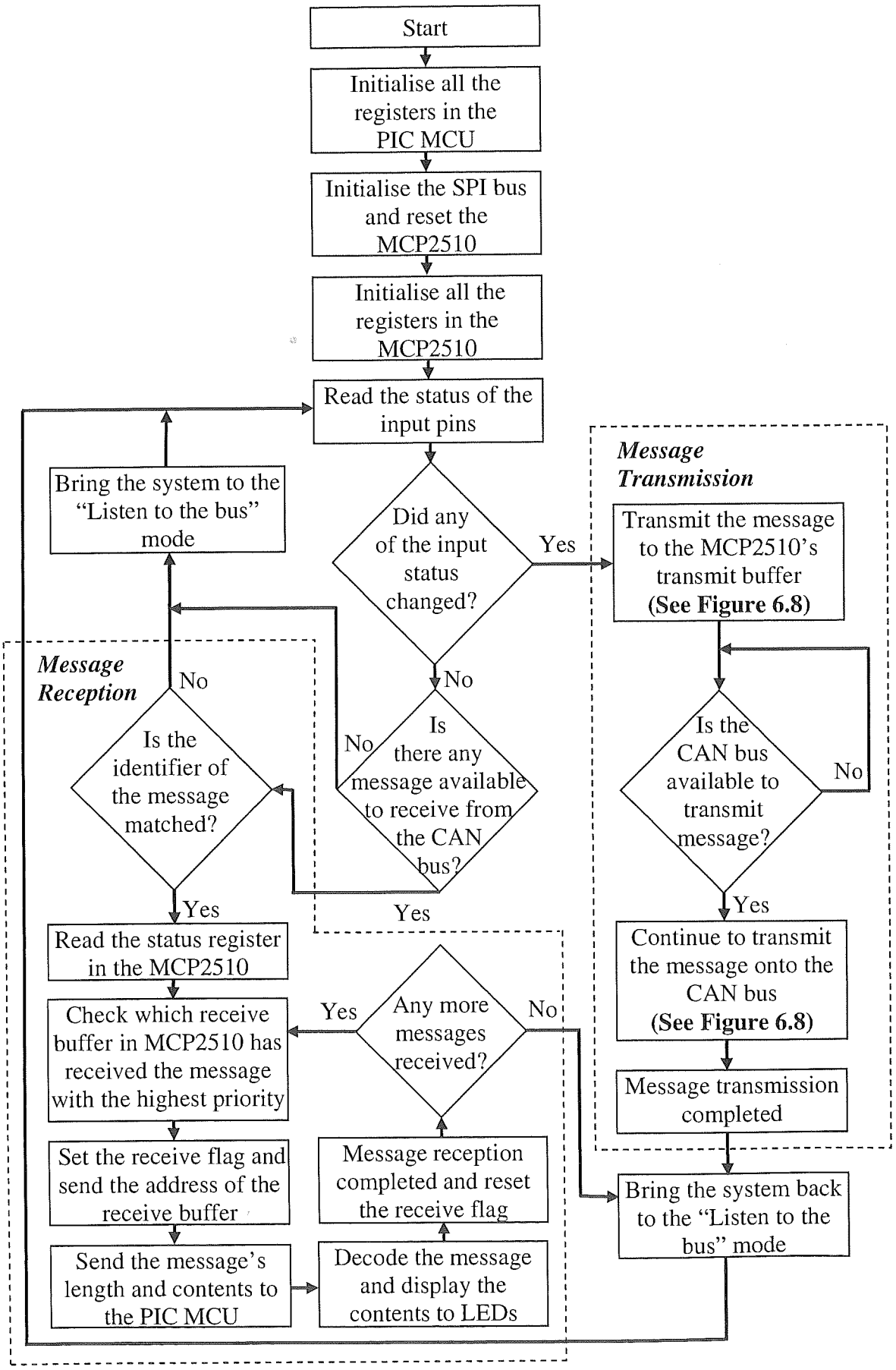


Figure 6.7 The main program flow of the software

After the initialisation process is completed, the PIC MCU starts to read the status of the input pins. When the information is received the PIC MCU checks to see if the received input status has changed from the previous setting. If the status has been changed, the program enters the "message transmission" loop to generate a CAN message according to the status and transmits it onto the CAN bus as shown in Figure 6.7. If the received status stays the same, the PIC MCU will then ask the MCP2510 to see if there is any message available to be received from the CAN bus. The program will enter the "message reception" loop if positive answer is received; otherwise the MCU will bring the system to the "Listen to the bus" mode. In the "Listen to the bus" mode, the system does not just constantly monitoring for any input status changes but also is prepared to receive messages from the CAN bus. The following sections presented the details of the message reception and transmission loops in the main program.

6.4.1 Message Reception

The process of the CAN message reception consists of eight steps as shown in Figure 6.7. As mentioned in Section 5.4.2, each message received is initially loaded into the MCP2510's MAB buffer and can only be transferred to one of the receive buffers in MCP2510 if the identifier of the message is matched to the filter. The PIC MCU then starts to process the messages that have arrived in the receive buffers. Only one received message can be processed each time the PIC MCU runs the loop. Messages with higher priority will be processed first and the PIC MCU will continue to run the loop until all the messages received in the receive buffers are processed completely.

In the message reception loop, after the received message is checked with the filter, the MCU will read the status register to see if the message has been accepted and stored in one of the receive buffers. Note that if more than one message is received, the information will be recorded in the status register which enables multiple looping. The next step of the process is to set the receive flag and send the register address information back to the PIC MCU for message reception. After the message is received, the PIC MCU will display the message contents to the output LEDs and reset the receive flag. To complete the process, the PIC MCU will check the status register again to see if there is anymore received messages in the receive buffer, otherwise, the system will return to the "Listen to the bus" mode.

6.4.2 Message Transmission

Figure 6.8 shows the details of the message transmission loop as illustrated in Figure 6.7. The CAN message transmission is divided into two stages:

1. Generates the message in CAN format according to the status of the PIC MCU's input pins and gets ready for the transmission onto the CAN bus by temporarily storing the message in one of the three transmit buffers in MCP2510.
2. Once the CAN bus is available for the message transmission, the message is then transmitted onto the CAN bus from the buffer and the system goes back to the "Listen to the bus" mode.

The first step is to enable the SPI bus and send the "Write" instruction to the MCP2510's transmit register followed by the address of the transmit buffer that the generated message that is wished to be sent is stored in. After the buffer's address is sent, the PIC MCU is ready to send the message length and followed by its contents to the buffer. Before transmitting the message onto the CAN bus, the system is required to check if the CAN bus is available for the transmission. If not, the process will be stopped temporarily, otherwise, the transmission continuous.

The process of continuing to transmit the message onto the CAN bus is identical to the procedure of sending the message to the transmit buffer via the SPI bus as described above. The "Write" instruction is first set in the control register, then the address of the register followed by the message are transmitted onto the CAN bus afterwards. The system will then go back to the "Listen to the bus" mode once the transmission is completed.

The overall message transmission process can be verified by oscilloscopes. Chapter 7 presents the digital waveform during the message transmission and the further discussions are also included in the chapter.

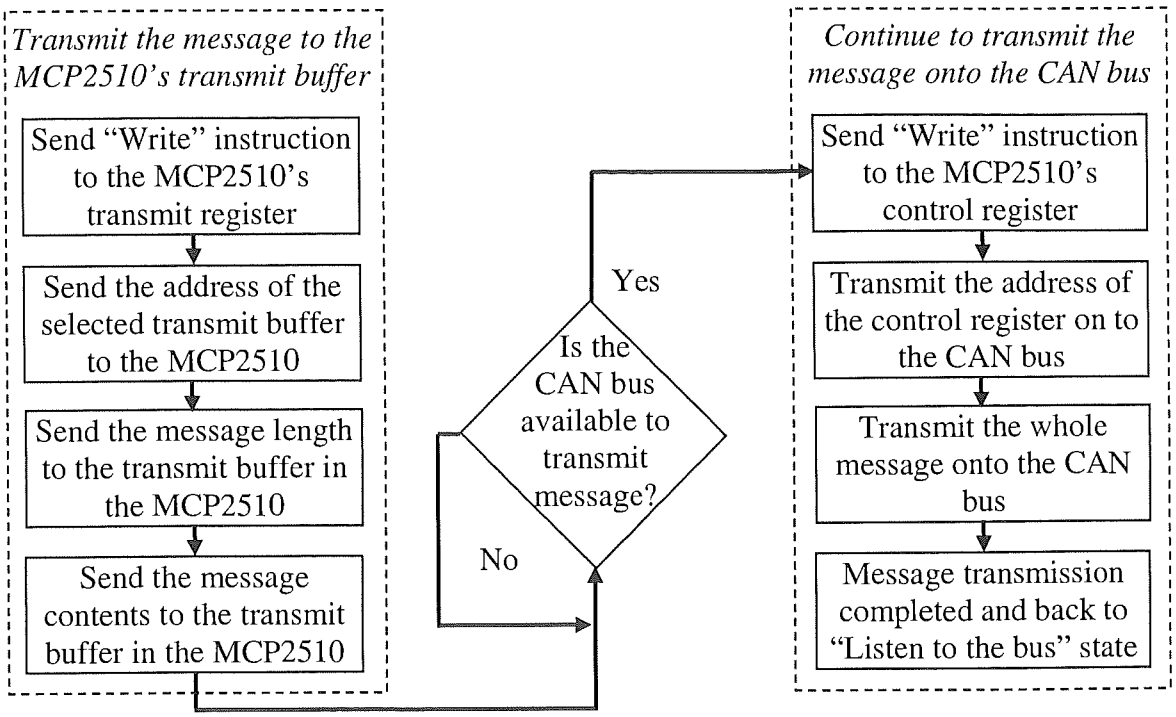


Figure 6.8 CAN Message Transmission

6.5 Summary

The CAN Control System is developed and implemented to turn on and off and detect open-circuit faults in the auxiliary loads of the MR2. The system hardware is broken down into three components, the CAN Control Node, the CAN Load Node and the Load Fault-Detector.

The CAN Control Node has the ability of monitoring eight toggle switches and automatically generating CAN messages based on the status of the switches. These messages will then be transmitted to the CAN Load Nodes via the CAN bus. The CAN Control Node is also capable of controlling eight LEDs, responding to message requests from the CAN Load Nodes via the CAN bus and generating time-based messages to turn on and off the Load Fault LEDs.

The CAN Load Nodes have a similar configuration and function to the CAN Control Node. The only differences are in the choice of MCU chipset and the circuitry that they are designed to interface to. Instead of reading signals from toggle switches, this board

is designed to read the signals from the Load Fault-Detector Board and generate CAN messages regarding to the signal received.

The Load Fault-Detector Board is implemented to turn on/off the 12V auxiliary loads. An additional feature of the board is to generate a +5V analogue signal once an open-circuit load fault is detected.

The software for the CAN Control Node and the CAN Load Node are written in C language. There are two main loops in the main program after the initialisation of the hardware: the message transmission and the message reception loops. The program will continues to run these loops until all the messages in either the transmit or receive buffers are fully processed.

Prototype of these boards are developed and constructed successfully. The next step of the project is to test these prototypes and evaluate their performance. All the details of the performance testing are presented in the following chapter, Chapter 7.

7 PROTOTYPE PERFORMANCE

Experimental tests are carried out to evaluate the performance of the Auxiliary Power System and the CAN Control System as the Power Distribution Network for the MR2. A 360W Cuk converter is used as the prototype of the Auxiliary Power System. This chapter presented the results of tests, under full load conditions; confirm the correct operation of the converter. Waveforms were captured during the tests and they are presented in the following sections of the chapter. The CAN Control System is also tested for the correct operation. Finally, the 360W Cuk converter prototype is combined with the CAN Control System hardware for overall system tests. In these tests, eight 12V, 45W light bulbs are used to act as the various loads in the MR2.

7.1 Converter Tests

The prototype of the 12V, 30A Cuk converter has been tested for efficiency and steady state operation. Under full load, the average current flowing through the Schottky diode in the converter is 30A. The diode's forward voltage of 0.86V limits the converter's theoretical maximum efficiency to 93.3%.

The measured efficiency of the switching converter at various loads is shown in Figure 7.1. From the figure, it appears that the highest efficiency of 92.4% is obtained at 2/3 load (I_{out} = 20A). Under full load condition (I_{out} = 30A), the efficiency is measured of 85.5%, which is above the minimum efficiency requirement of 85% as listed in Table 3.2. However, according to the simulation results as illustrated in Figure 3.5, under full load condition, the maximum efficiency at the 100 kHz switching frequency is 90.2%. This shows that there is a 5% difference in efficiency between the simulation and the experimental results. The 18W (5% of 360W) of power loss is mainly due to the extra

losses (or heat dissipation) on the switching devices during the experiments, which were not considered when the simulations were carried out.

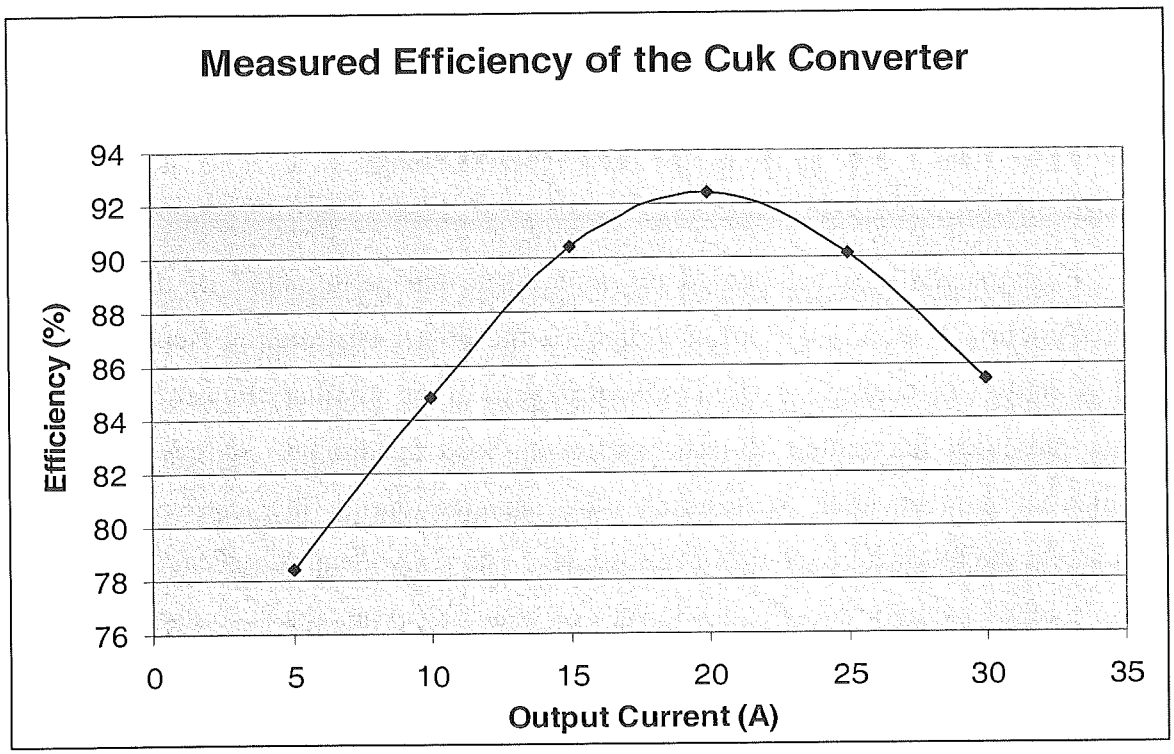


Figure 7.1 Measured efficiency of the 360W Cuk converter prototype

The characteristic switching waveforms of inductor L1, MOSFET and diode are shown in Figure 7.2 and Figure 7.3. These figures show reasonably clean switching under the full load condition and confirm the correct operation of the converter. Ideally, the nominal voltage across the power devices (MOSFET and Diode) is 60Vp-p. As seen in Figure 7.2 and Figure 7.3, voltage spikes occur when the switches are turned on or off. Voltage clamps (RCD Snubbers) are designed and implemented across these power devices to reduce the voltage spikes and to limit the overall peak-to-peak voltage. Consequently, the maximum voltage across the MOSFET and diode is 95Vp-p and 92Vp-p respectively when using the voltage clamps. The current ripple of the inductor L1 and output voltage ripple (as shown in Figure 7.4) under full load condition are measured as 514mAp-p and 168mVp-p respectively. These values are close to the simulation results of 500mAp-p and 140mVp-p as given in Section 3.4.

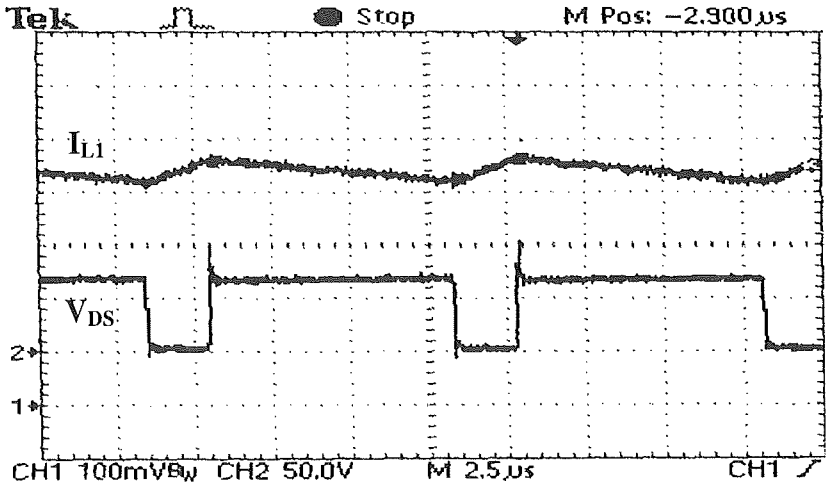


Figure 7.2 Input inductor current ripple (Top, 2A/div), and MOSFET switch voltage waveform (Bottom, 50V/div). Time scale: 2.5µs/div

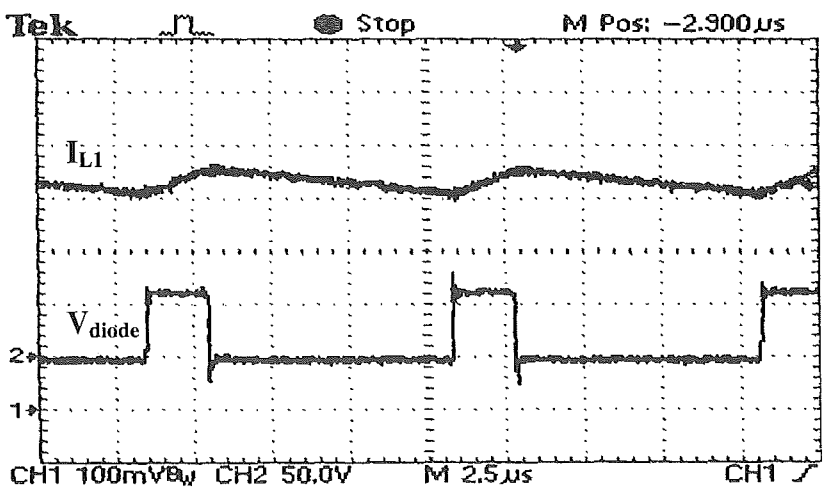


Figure 7.3 Input inductor current ripple (Top, 2A/div), and Diode switch voltage waveform (Bottom, 50V/div). Time scale: 2.5µs/div

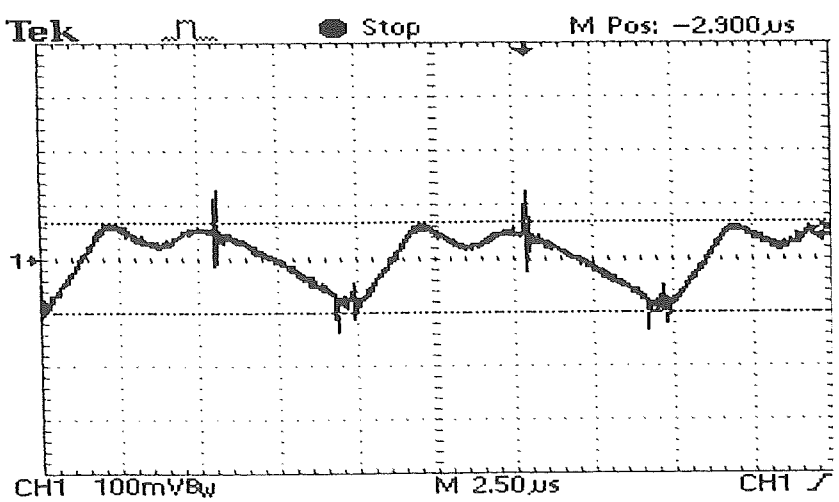


Figure 7.4 12V output voltage ripple (100mV/div). Time scale: 2.5µs/div

7.2 Converter Control Tests

In order to test the voltage feedback compensator implemented in the converter, two tests are carried out: the output voltage stabilising test and the transient loading response from no load to full load condition. Figure 7.5 shows the graph of the measured output voltage when the output current increased from 0 to 30A. From the graph, it appears that the measured output voltage decreases linearly with respect to the increase of loading. The voltage difference between no load and full load condition is measured of 0.36V. This gives the converter's output voltage rating of 12V ± 2%, which is within the specification of ± 10% listed in Table 3.2.

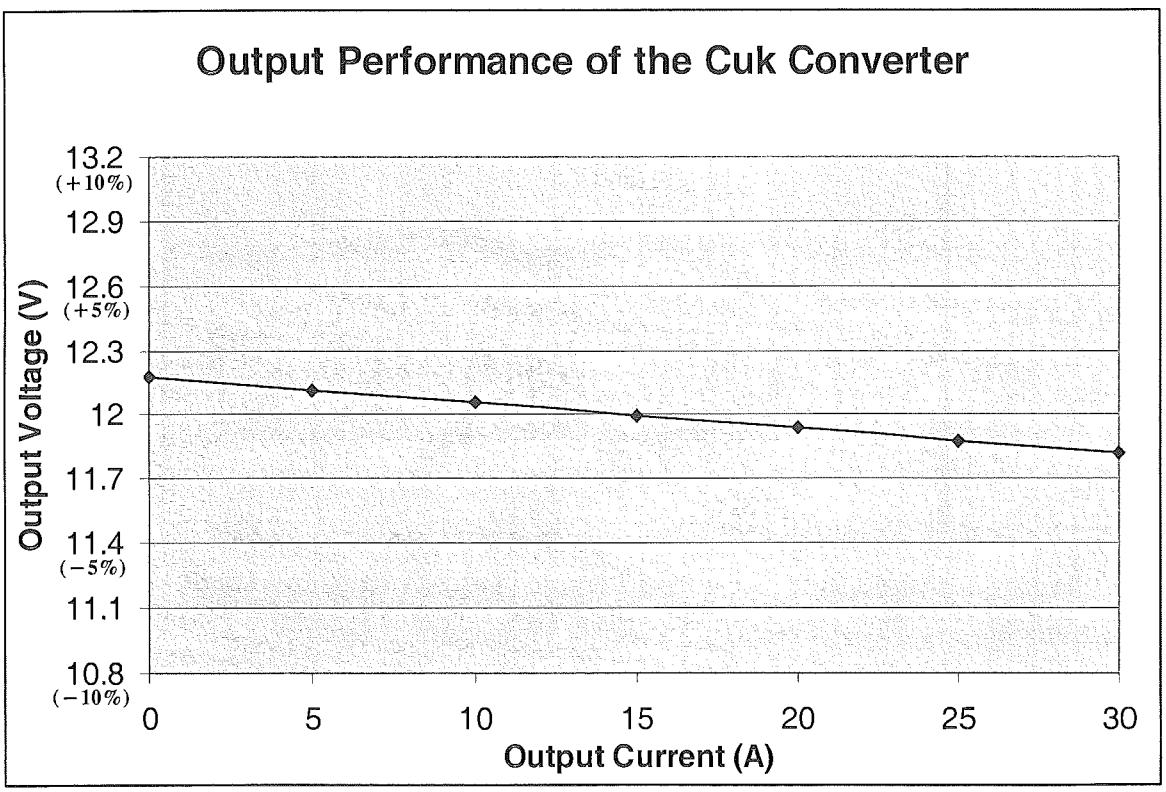


Figure 7.5 12V output voltage characteristic

A transient response test of the converter is considered important as it determines the converter's output voltage recovery time after the load is changed. As mentioned in Section 4.4, the converter's response time is designed to be 2ms (500Hz). Experimental equipment was set-up and Figure 7.6 shows the transient response of the converter for a load change from 0 to 30A. From the figure, it appears that the measured response time is

60ms (12.5Hz), which is longer than the designed value of 2ms (500Hz). The output voltage has also dropped significantly to a value, as low as 3V during the test. This is because the converter's control loop is over-damped and the delay is mainly caused by the current limiting control circuit limiting the current spike to the 33A boundary. This delay time can be reduced by increasing the converter's response frequency above 500Hz, however with the risk of un-stabilising the system. Since majority of the car applications do not require the full current rating and that the 60ms of delay time is hardly noticeable with the human eye for lighting applications, the result of the converter's transient response under full load condition is acceptable.

Figure 7.7 shows the test result when current limiting the converter's output to 30A. The output voltage starts to drop significantly as the output current exceeds 30A. The shutdown protection then activates when the output current finally reaches 33A boundary. Figure 7.8 provides a detailed look at the instant the shutdown circuit activates and the measured time delay is 0.6ms. These results show that the protection circuits that have been designed and implemented in the converter are working successfully.

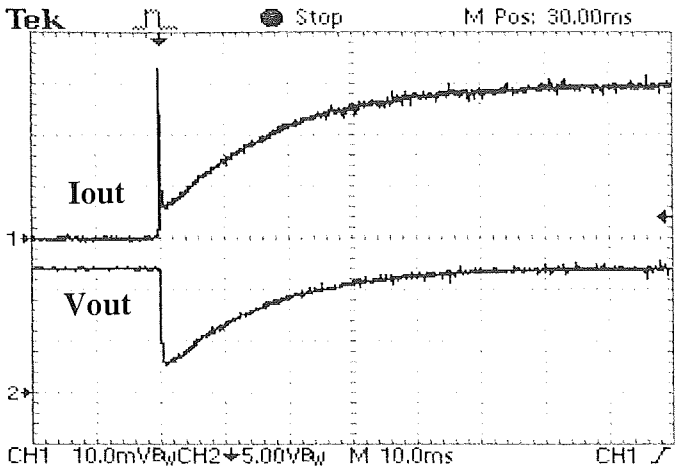


Figure 7.6 Transient loading respond under full load, Time scale: 10ms/div (Top: Output Current @ 10A/div, Bottom: Output Voltage @ 5V/div)

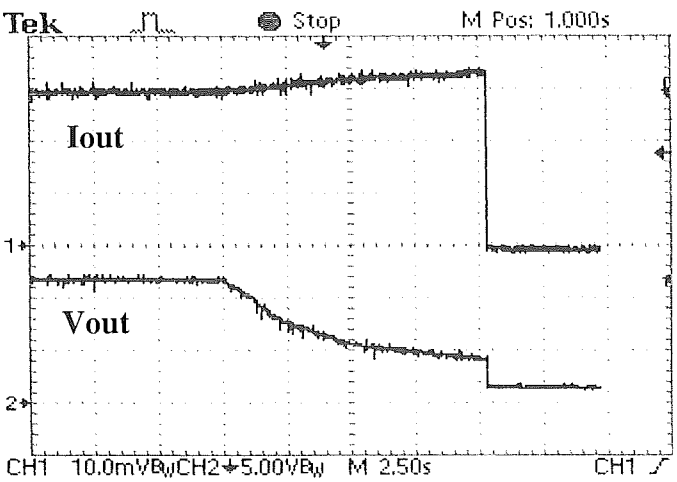


Figure 7.7 Current limiting and shut-down at full load, Time scale: 2.5/div (Top: Output Current @ 10A/div, Bottom: Output Voltage @ 5V/div)

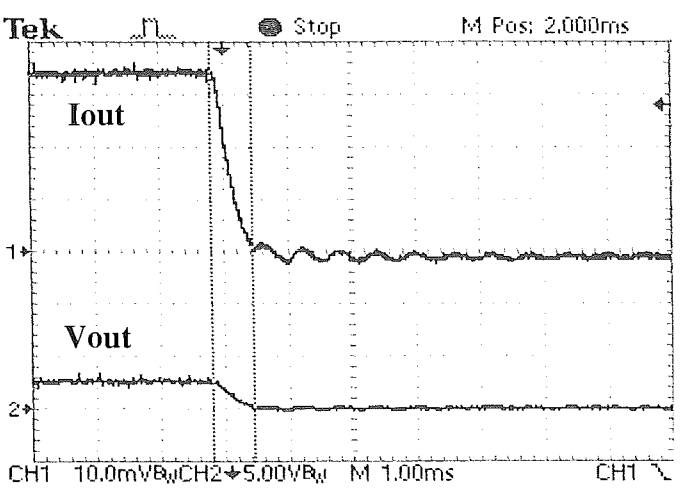


Figure 7.8 A close look at the shut-down operation, Time scale: 1ms/div (Top: Output Current @ 10A/div, Bottom: Output Voltage @ 5V/div)

7.3 CAN Control System Tests

The CAN Control System hardware is constructed and tests are carried out to ensure the correct operation. Message waveforms at the SPI bus are captured to verify that the messages are sent from the PIC MCU and received at the MCP2510. The messages are then transmitted onto the CAN bus and waveforms are recorded. These waveforms provide the time delay between signalling to turn on a load and the current flowing in the load.

Figure 7.9 shows the message transmission from the PIC MCU to the CAN bus via the SPI bus and the MCP2510 CAN controller. There are 4 signals recorded in the figure: Analogue switch (with turn-off operation), CS (Chip Select), SCLK (Signal CLocK), and the SI (Signal Input) of the MCP2510.

As mentioned in Section 6.4.2, the message transmission consists of 2 stages: firstly, writing and storing the message in one of the three transmit buffers in MCP2510 and then transmits the message onto the CAN bus from the buffer. Figure 7.9(a) and (b) shows the message transmission from the PIC MCU to the MCP2510 via the SPI bus. The message transmission is started by lowering the CS Pin as shown in Figure 7.9(a). The first byte of data transmitted is the "write" instruction. After the write instruction is sent to the MCP2510, the address of the selected transmit buffer is sent. In this case, transmit buffer 0 is selected to temporarily store the message. Then the length of the message is transmitted followed a single byte of message which contains a value of 0 since the switch is turned-off, as is shown in Figure 7.9(b). After the message is transmitted and stored in the MCP2510, the message now is ready to be transmitted onto the CAN bus. Figure 7.9(c) shows the write instruction is sent followed by the address of the control register. The MCP2510 then transmits the message onto the CAN bus once the "request to send" instruction is confirmed and sent to the MCP2510. After that, the MCP2510 goes back to the "listen to the bus" mode once the message transmission is completed.

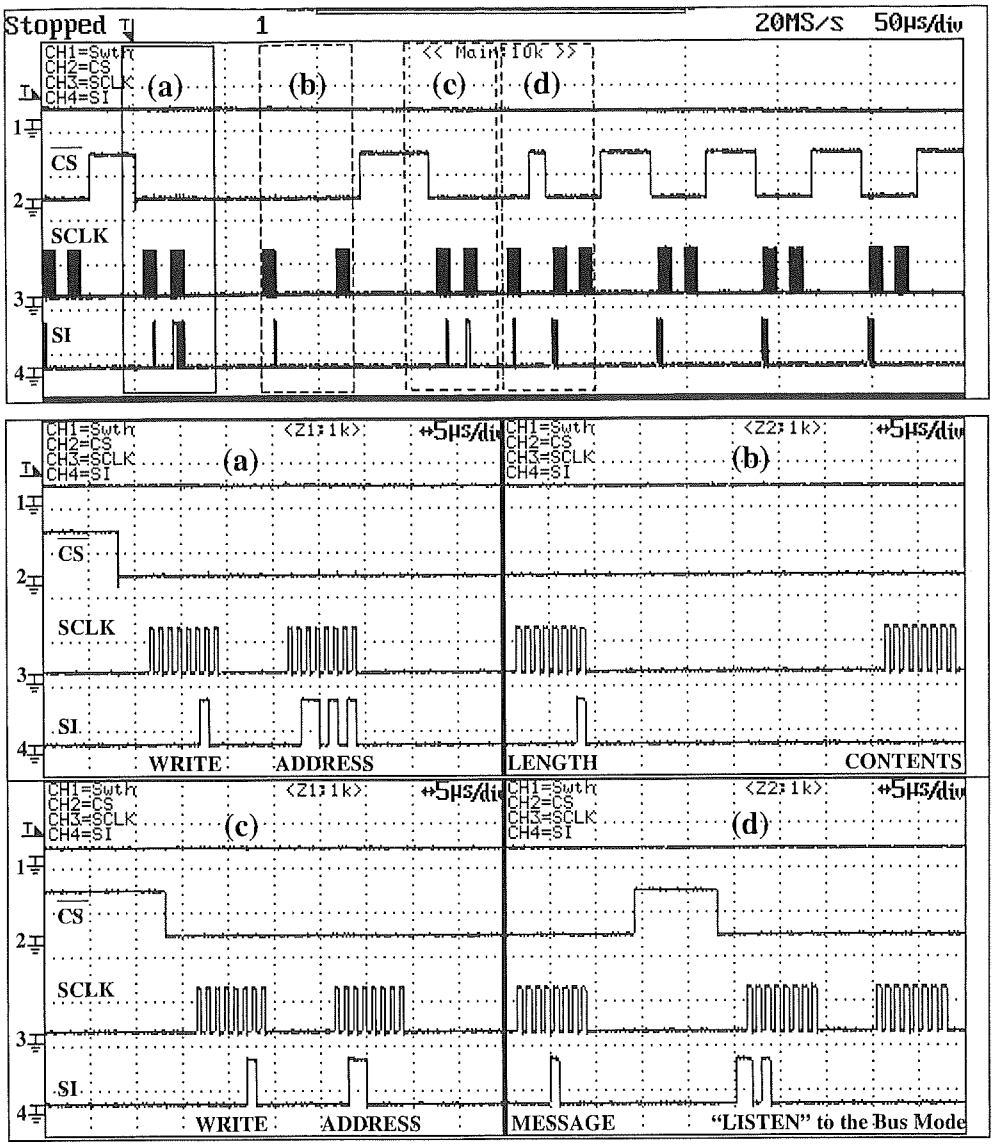


Figure 7.9 Message transmission on SPI bus to the MCP2510 divided into: (a) Set "write" instruction and the transmit buffer address in MCP2510. (b) Send the message length and message to the transmit buffer. (c) Prepare to transmit the message onto the CAN bus by following the steps in (a). (d) Send the instruction to place the message onto the CAN bus and then tell the MCP2510 to go back into the "listen to the bus" mode once the transmission is completed.

Figure 7.10 shows the waveform captured when transmitting the "turn on light bulb numbered 6" message between the switch and the load via the CAN bus. The time taken to transmit the message from the switch to the CAN bus is 362µs and the overall delay between the switch operating and the load turning-on is 1.04ms. This indicates that the switching response of the loads via the CAN bus is relatively fast for the MR2 application. The time delay between the switch signal and the load switching off (1.25ms) is found longer than when it switches on. The message transmission delay for turning off light bulb numbered 6 is shown in Figure 7.11.

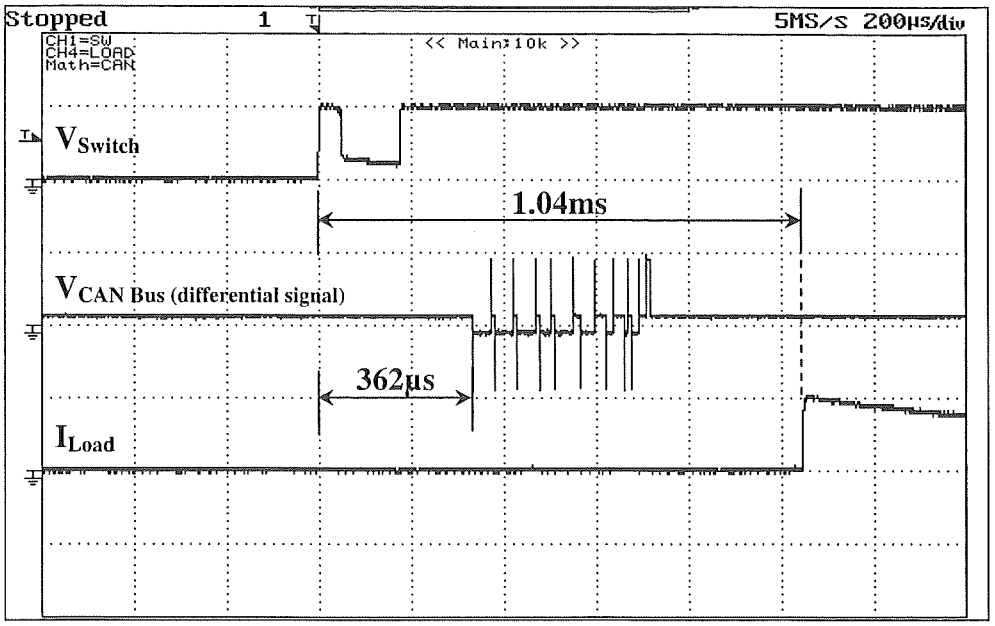


Figure 7.10 Transmission delay between the switch on signal and the load switching on. Time scale: 200µs/div. (Top: Voltage across switch @ 5V/div; Middle: Differential voltage across the CAN bus @ 5V/div; Bottom: Current flow through the light bulb @ 5A/div)

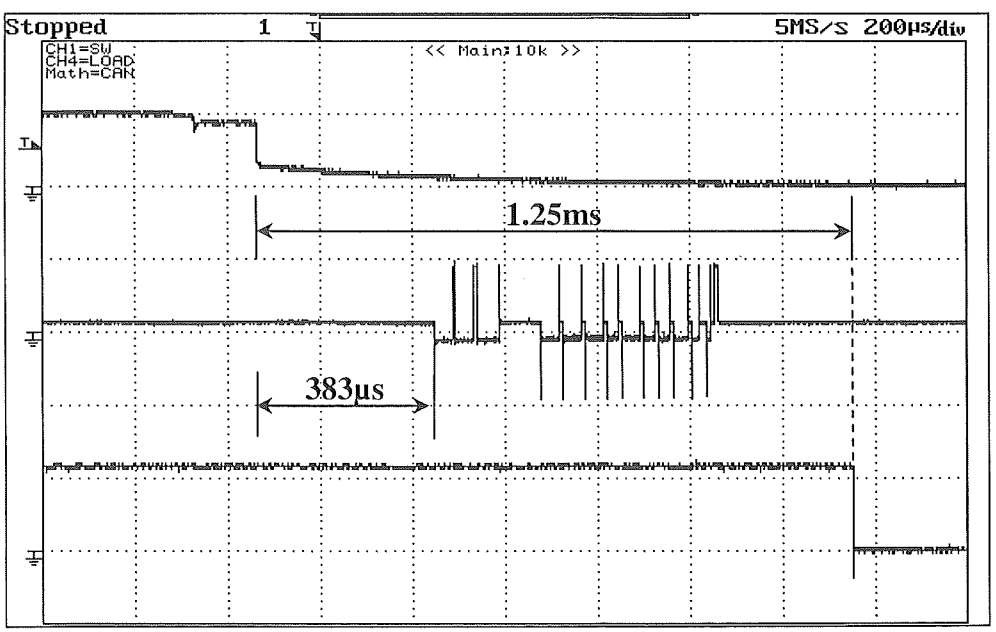


Figure 7.11 Transmission delay between the switch off signal and the load switching off. Time scale: 200µs/div. (Top: Voltage across switch @ 5V/div; Middle: Differential voltage across the CAN bus @ 5V/div; Bottom: Current flow through the light bulb @ 5A/div)

7.4 Power Distribution Network (Overall) Tests

The overall Power Distribution Network is tested by connecting together the Cuk converter and the CAN Control System. There are eight toggle switches mounted on the CAN Control Node board to turn on and off the eight 45W, 12V light bulbs. In the test, all the light bulbs are to be switched on and off in a sequence and the measured current waveform from the converter is captured in Figure 7.12.

Figure 7.12(a) shows the output voltage and current waveforms captured when switching on all the light bulbs in succession. From the figure, it can be seen that the current spikes occur when the lights are initially turned on. However, the current spikes of the last two light bulbs are reduced significantly as the 30A current limit is reached. This indicates that the current spike is eliminated successfully by the current limiting control. The waveform captured at the output when turning off the light bulbs in succession is shown in Figure 7.12(b).

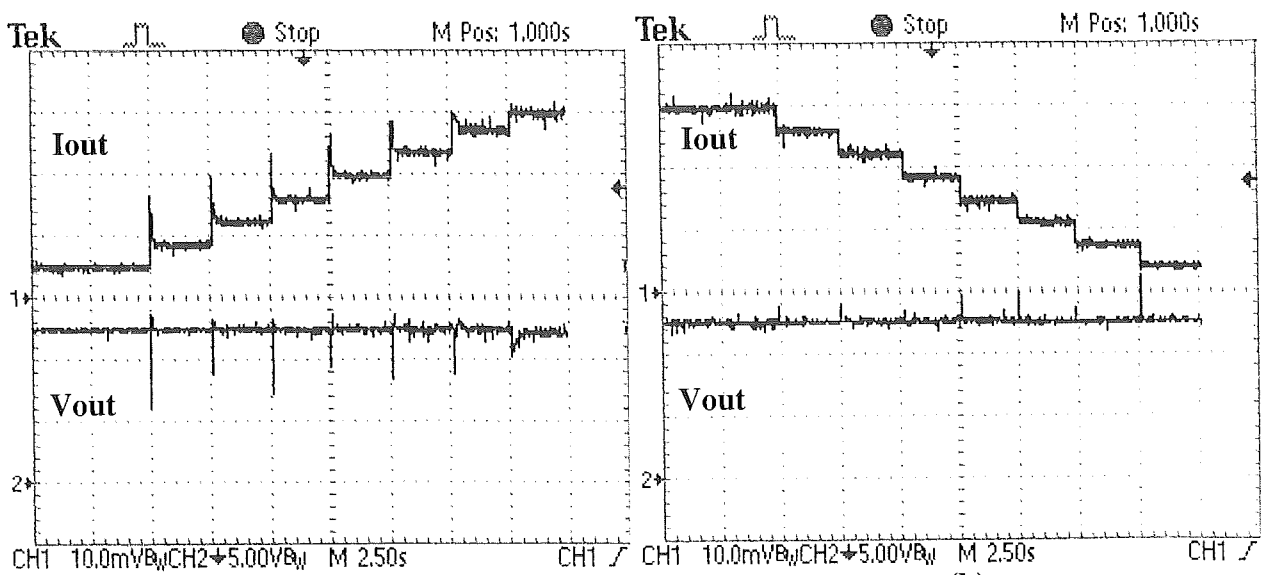


Figure 7.12 Waveform captured at the output when:
(a) Turning on all the 12V, 45W light bulbs (Time scale: 2.5s/div)
(Top: Output Current 10A/div, Bottom: Output Voltage: 12V/div)
(b) Turning off all the 12V, 45W light bulbs (Time scale: 2.5s/div)
(Top: Output Current 10A/div, Bottom: Output Voltage: 12V/div)

7.5 Prototype Photographs

Photographs were taken during the testing of the Power Distribution Network prototype. Figure 7.13 shows the photograph of the testing environment of the prototype. Three of 120W bench top DC power supplies are used to provide the input power to the Cuk converter. Measurement equipment such as oscilloscopes, voltmeters and ammeter are used during the tests. The CAN Control System boards are powered by the Cuk converter and controls the eight of 45W, 12V light bulbs as shown in the figure.

Due to the limitation of the photograph size, some details of the 360W Cuk converter and the CAN Control System hardware can not be shown clearly. Therefore, extra photographs are provided as shown in Figure 7.14 and Figure 7.15 to take a close look at these system prototypes. Figure 7.15 shows how the CAN Control System boards are joined together with the 3 meter CAN bus. The CAN bus is placed around the system boards to monitor the noisy environment inside the MR2 during the tests.

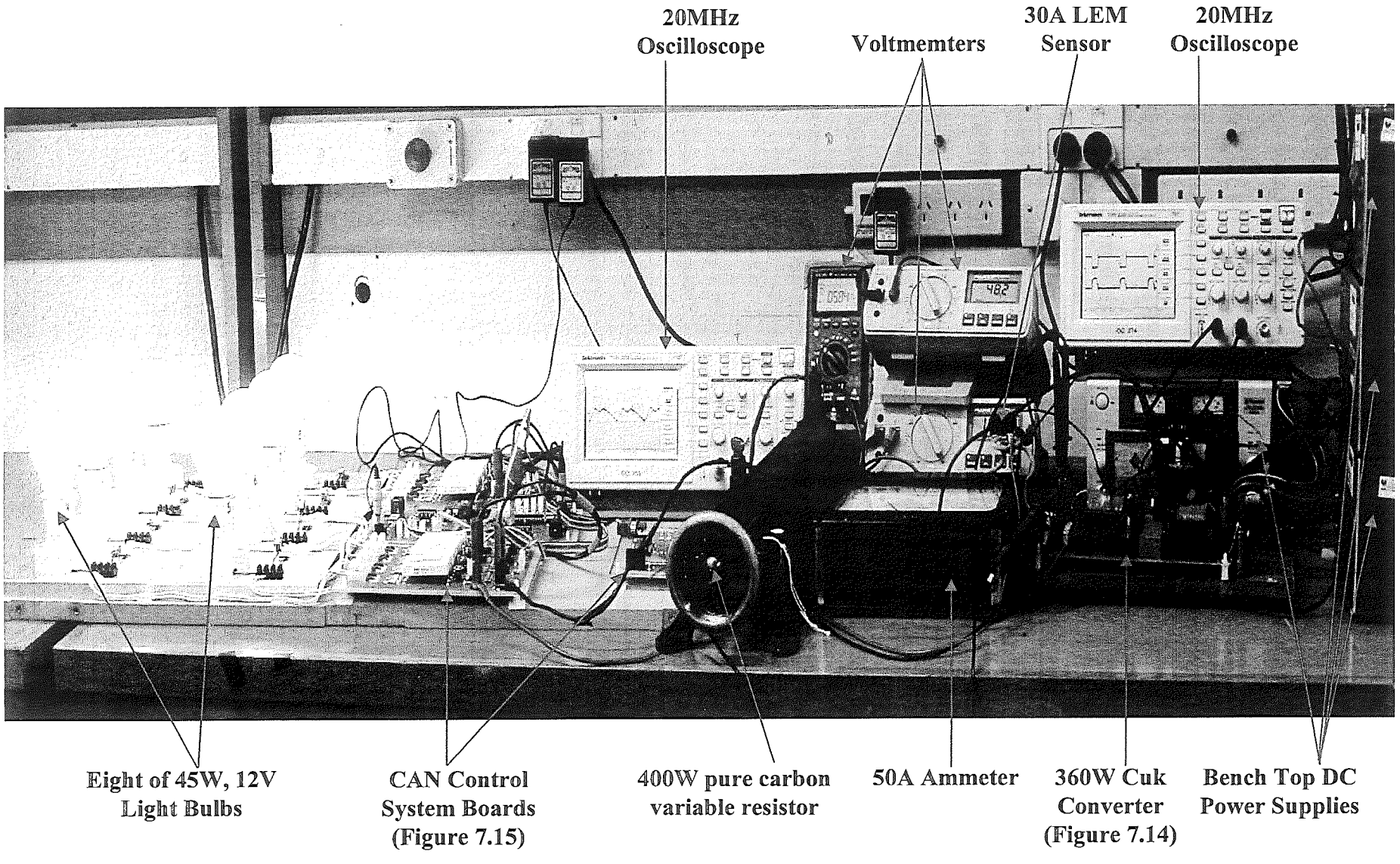


Figure 7.13 Power Distribution Network (Overall system prototype) testing environment photograph

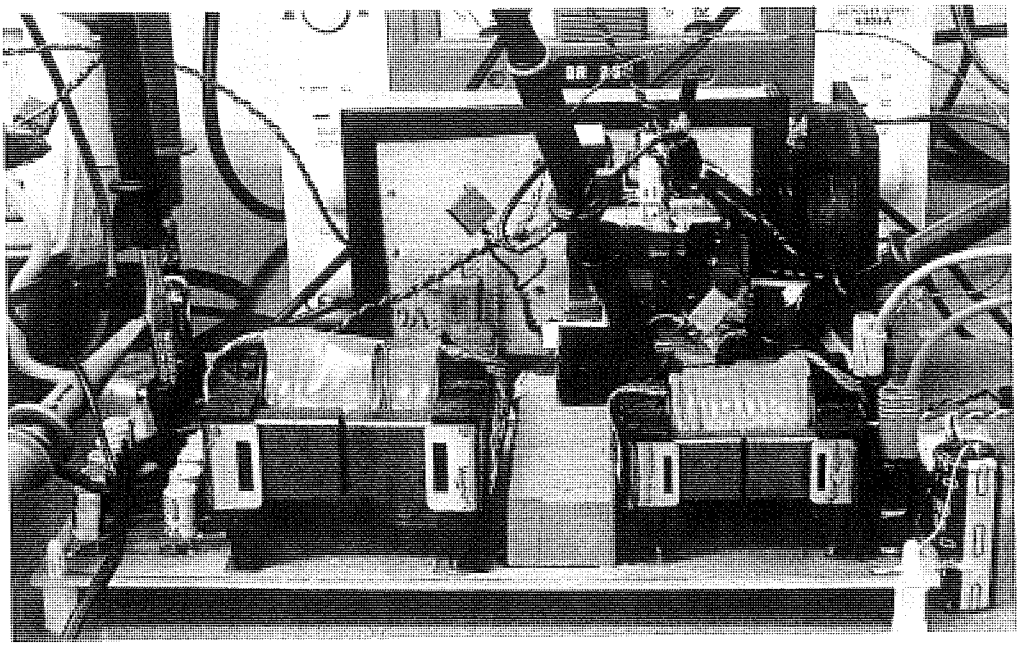


Figure 7.14 360W Cuk converter (the Auxiliary Power System) testing photograph

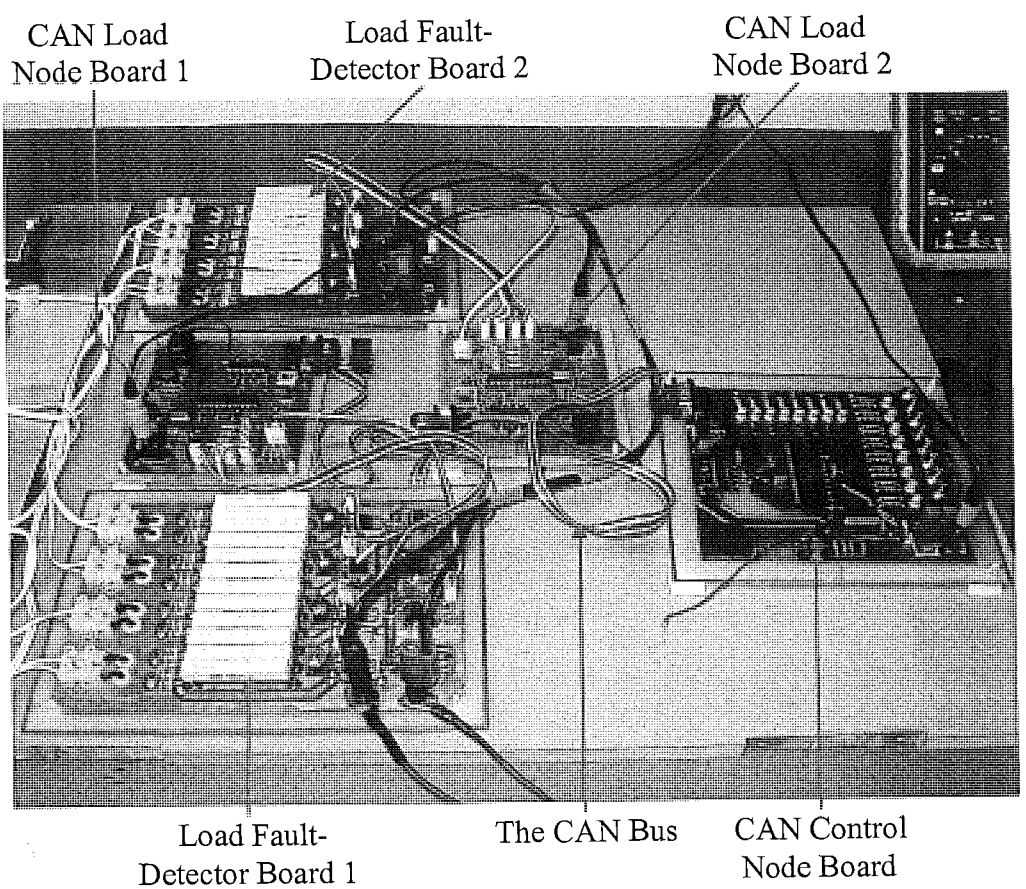


Figure 7.15 CAN Control System prototype testing photograph

7.6 Summary

The prototype of the Power Distribution Network was constructed and tested successfully. The 360W Cuk converter operates at 100 kHz and has a maximum efficiency of 92.4% at 2/3 load. Under the full load condition, the switching waveforms of the switching devices were captured to confirm the correct operation of the converter. The measured current ripple and the voltage ripple match the simulation results presented in Section 3.4.

The current-mode control circuits are also tested with the converter. The measured converter's output voltage rating is 12V ± 2%, which is within the design specification. The transient response of the converter under full load condition is measured 60ms (12.5Hz), which is longer than the designed value of 2ms (500Hz). This is because the converter's control loop is over-damped and the delay is mainly caused by the current limiting control circuit. This delay time can be reduced by increasing the converter's response frequency above 500Hz, however with the risk of un-stabilising the system.

Tests on the CAN Control System hardware were also carried out. The digital signal waveform of message transmission on SPI bus to the MCP 2510 was captured to verify the correct operation of the system. The overall time delay of the message transmission via the CAN bus was found to be 1.04ms. This is considered to be fast enough for the MR2 application, with the driver not noticing the delay. Finally, the overall system was tested and all the 12V light bulb that are used to simulate the auxiliary loads in the vehicle were switched on and off successfully. From these experimental results, it can be concluded that the Power Distribution Network developed for the auxiliaries in the MR2 is expected to operate successfully.

8 FUTURE DEVELOPMENT & CONCLUSION

8.1 Future Development

A Power Distribution Network for the newest departmental electric vehicle, a Toyota MR2 has been designed, constructed and tested. The Power Distribution Network was developed to control and supply power to the auxiliary loads in the MR2. The network consists of two major systems: the Auxiliary Power System and the CAN Control System. A Cuk converter was constructed as the Auxiliary Power System. The converter operates off a nominal supply voltage of 48Vdc, and is able to drive up to 360W of 12Vdc auxiliary load. The next stage of the system development is to construct another three auxiliary power supplies with power ratings of 360W and install them into the MR2.

A CAN Control System prototype was also constructed to control the auxiliary loads using the CAN bus technology. The prototype consists of two analogue boards and three CAN node boards, and the operating software was written for these node boards. The purpose of implementing the system was to demonstrate the basic operation of CAN and use the concept in the MR2 application. The next stage of the system development is to modify some of the software so that the system can also be able to control the inductive type of auxiliary loads such as power steering and window wipers. Another two CAN Load Node and Load Fault-Detector boards are required to be constructed and installed into the vehicle.

Finally, high intensity LEDs are required to replace the existing external and internal light bulbs to further reduce the current demand of the loads. Future theoretical and experimental analysis are required to estimate how efficient and reliable is the Power Distribution Network is when it controls and supplies power to the auxiliary loads in the vehicle.

8.2 Conclusion

This thesis documented the design, implementation, and test results of a Power Distribution Network for the MR2 electric car application. It is proposed that the 300Vdc nominal battery voltage is converted to a 48Vdc intermediate voltage. The use of a 48Vdc distribution voltage is considered more efficient than the usual 12Vdc distribution system since smaller and lighter wires can be used to carry same amount of power.

Current consumption investigations were made on the MR2's auxiliary loads. In order to use the limited battery energy more efficiently, current limiting capability was added to the converters with the intention of reducing the peak current demand of the auxiliary loads.

The type of loads determined the power rating, number and placement of the auxiliary dc-dc converters that form the Auxiliary Power System in the vehicle. The Cuk converter was selected because it has both continuous input and output current with high efficiency. Theoretical calculations were made to estimate the component losses in the converter at the various switching frequencies. A switching frequency of 100 kHz was selected as the best compromise between the efficiency and the component size.

PWM and protection circuits are required to control and protect the converter. Current mode control circuitry was selected and the control circuit consists of three control paths: voltage compensation, pulsed-by-pulsed current limiting, and automatic shut-down protection. The voltage feedback compensation was designed with a bandwidth of 500Hz. The current limiting control was set to activate once the output current was exceed 30A. If the output current was exceeding 33A, the converter will be turned off by the automatic shut-down circuitry to protect the converter from the over-current damage.

The CAN topology is selected for communicating around the electric vehicle since the CAN protocol is optimised for systems that need to transmit and receive relatively small amount of information reliably to any or all other nodes on the network. Since the protocol is message-based, all nodes on the bus receive every message and acknowledge every message, regardless of whether the node required the data or not. Fast robust message transmission with fault confinement is the big plus for CAN because faulty nodes will

automatically drop off the bus, which does not allow any faulty node to bring down the network.

Various CAN implementation methods were studied and the stand-alone CAN controller configuration was chosen. This is because with this configuration, multiple designs could be developed in the future and the software and hardware development is reusable. The CAN Control System was designed and consists of five CAN nodes, based on the PIC16F87X MCU and MCP2510 CAN controller. Prototype of these boards are developed and constructed successfully.

The prototype of the Power Distribution Network was constructed and tested successfully. The 360W Cuk converter operates at 100 kHz and has a maximum efficiency of 92.4% at 2/3 load. Under the full load condition, the switching waveforms of the switching devices were captured. Tests on the CAN Control System hardware were also carried out. The digital message transmission on SPI bus to the MCP 2510 was captured to verify the correct operation of the system. The overall time delay of the message transmission via the CAN bus was found to be 1.04ms. This is considered to be fast enough for the MR2 application, with the driver not noticing the delay. Finally, the overall system was tested and all the light bulb loads that are used to simulate the auxiliary loads in the MR2 were switched on and off successfully.

Finally, the prototype Power Distribution Network that consists of a 360W Cuk converter and a CAN Control System has been successfully implemented. It is expected that a fully working Power Distribution Network will operate successfully in the MR2.

REFERENCES

[1] ABC of AFV's (1997), Electric Vehicle Histories, California Energy Commission, June 1997.

[2] Byers D.J (1983), "AC Induction Motor Drive Systems for Electric Vehicles" Transactions of the IPENZ vol.pp1-14

[3] Williams M.R. (1993), "A Controlled Current Inverter for an Electric Vehicle", Master Thesis, University of Canterbury.

[4] Xiaoming Li, (1999), "Design of a Three-Level inverter for an Electric Vehicle", Master Thesis, University of Canterbury.

[5] Electrical and Computer Engineering Department, University of Canterbury Website, www.elec.canterbury.ac.nz.

[6] T. S. Finnegan, "Cuk: the best SMPS?", Electronics World + Wireless World, January 1991, pp 69-72

[7] S. Cuk and R.D. Middlebrook, "A New Optimum Topology Switching dc-to-dc Converter", IEEE Power Electronics Specialist Conference 1977 pp 160-179

[8] Zhe Zhange and S. Cuk, "A High efficiency 1.8kW Battery Equalizer" Applied Power Electronics Conference, 1993 Record, pp.221-227.

[9] Daniel W. Hart, *An Introduction to Power Electronics*, Upper Saddle River, N.J. : Prentice Hall, c1997, pp.205-201.

[10] Keith Billings, *Switch Mode Power Supply Handbook*, 2nd Ed, McGraw-Hill, Inc, c1999, pp.3.3-3.28.

[11] Philips Soft Ferrites Data Book 1998.

[12] Mammano Robert, Switching Power Supply Topology Voltage Mode vs. Current Mode, Unitrode Design Note, Oct 1994.

[13] Bosch GmbH, CAN specification, Version 2.0, September 1991.

[14] Keith Pazul, Microchip Technology Inc., AN713, Controller Area Network (CAN) Basics, May 1999.

[15] Microchip Technology Inc., MCP2510 data sheets, Stand-Alone CAN Controller with SPI™ Interface, June 1999.

APPENDIX

Table of Contents

APPENDIX A1

MR2's Auxiliary Current Consumption Table

Current Usage for a Toyota MR2						
Load	**Description**	**Current Measurement (Amps)**		**Time To Reach Steady State**	**Comments**	**References**
		Initial Peak	**Steady State**			
Front Lights	Cap Rise	22A	5A, 0A	250ms, 540ms	No Head Lights Operating	
	Cap Down	21.2A	3.2A, 0A	250ms, 530ms		
	Low Beam Headlights	27.4A	8.2A	500ms	No Cap Operating	
	High Beam Headlights	27.4A	9.8A	500ms		
	Emergently Headlights On	27.4A	10.6A	600ms	Cap Rise + High Beam Headlights	
	Front Fog Lights	*Not working*				
	Hazard Lights	27.4A	8A, 0A	250ms, 400ms	6 Indicator Lights Operating	
Back Lights	Dim Lights	26.4A	5A	75ms	2 Yellow Front and 4 Red Back Lights	
	Break Lights	27.4A	7.4A	250ms	4 Red Break Lights Operating	
	Back Up Lights	*Can not make it work*				
	Right Handed Indicators	18.6A	4A, 0A	200ms, 250ms	3 Indicator Lights Operating	
	Left Handed Indicators	*Exactly same results obtained as above*				
Internal (Dome) Lights	Driver Side Reading Light	4.4A	0.64A	75ms	1 Bright yellow Light Operating Only	
	Both Reading Lights	5.32A	1.2A	75ms	2 Reading Lights Operating	
	Door Lights and Sound	4.24A	0.64A	75ms	Sound Is Based On The Key Position	
	Door and Reading Lights	8.08A	1.76A	100ms	2 Reading Lights; 2 Red Door Lights	
	Back Door Light	2.24A	0.28A	75ms	1 Bright Yellow Light Operating	
Heater Fan	Shift Car Key To The ACC Position	3.52A	3.52A, 2.6A, 1.84A	280ms, 2s, 6.36s	Key Shift To The ACC Position Is Required Before Operating The Heater	
	On Low	5.4A	3.8A	1.5s	All Been Operated From Off; No Change On The Current Value When Alternate The Directions Of Air-Blow	
	On Medium	15.6A	7.4A	1.1A		
	On High	25.6A	11.8A	750ms		
	Low to Medium to High	5.4A, 10.2A, 18A	3.8A, 7.6A, 11.2A	N/A	To Investigate The Sparks Formed; Measurements Were Done Manually	

Figure A. Current Consumption Table 1 of 2

Load	Description	Current Measurement (Amps)		Time To Reach Steady State	Comments	References
		Initial Peak	Steady State			
Power Steering	Turn On	49.2A	12A	2s	While No Movement On the Wheels	
	Turn To Right Hand Ended	69.6A	56.4A	N/A	Hard Lock On the Steering, These Were Done Manually, So No S.S. Avaliable.	
	Turn To Left Hand Ended	*Same As Results Obtained Above*				
	Turn From Right Hand End To The Left Hand End	66A	56.8A	N/A	Turn From Right Hard Lock To The Left Hard Lock	
Power Window	Open Driver Side Window	16.8A	2A, 15.6A	180ms, 2.8s	Use Driver Side Automatic Window Winder For Open And Close Window	
	Close Driver Side Window	17.4A	4.8A, 16.4A	3.65s		
	Open Both Window	23.6A	4.6A	200ms	These Were Done Manually, Therefore No S.S. Available At The End.	
	Close Both Window	23.6A	9.8A	220ms		
Window Wiper	Intermit Speed	2.22A	N/A	N/A	All Been Operated From Off; No S.S. Available But Has **Average Current of 0.5A For Low & 1A For High Speed**	
	Low Speed	2.04A	N/A	N/A		
	High Speed	2.16A	N/A	N/A		
	From Low To High Speed	1.56A	N/A	N/A	For Sparks Investigation Purpose. No Sparks Presented On the Curve	
	From High To Low Speed	1.62A	N/A	N/A		
Car Audio	Radio On With Aerial Up	5.68A	2A, 1A	1s, 5s	No Significant Change On Current Readings While Alternate Volumes	
	Radio Off With Aerial Off	5.12A	1A, 0A	2s, 4.5s		
	Tape Player	*While The Radio Was On, No Change On The Current Readings When Insert A Tape*				
	CD Player	*Automatically Turned On While At ACC Stage, Please Refer To ACC Measurement*				
Central Locking	Lock Doors	9.12A	8.24A, 0A	100ms, 1ms		
	Unlock Doors	8.88A	7.6A, 0A	150ms, 1ms		
Other Electronic	Horn Operation	10A	N/A	N/A	PWM Waveform Obtained, **Average Current of 10A**	
	Wing Mirrors Shifting	0.436A	0.1A	100ms	Can't Make Mirror-Exacting Work	
	Rear Defogger On	3.92A	3.8A	250ms		
	Air Conditioning	*Air Conditioning does not work*				
	Combination Meters	4.2A	3A	100ms	Only An Estimate Result	

Figure B. Current Consumption Table 2 of 2

APPENDIX B1

Component Calculation of the Cuk Converter

A C'uk converter has an input of ***48Vdc*** and is to have output ***12Vdc*** supplying a ***360W*** load (note: this is because 12Vdc*30A$_{limit}$=360W). Switching frequency is ***100kHz***; with the high efficiency ***100%*** must be obtained. The Requirement of inductor sizes such that the change in inductor currents is no more than ***10%*** of the average inductor current, the output ripple voltage is no more than ***1%***, and the ripple voltage across capacitor is no more than ***5%***.

Solution:

The duty ratio is: [Equation (1) applied]

$$\frac{Vo}{Vs} = -\frac{D}{1-D} = \frac{12}{48} = 0.25$$

$$\mathbf{D=0.33}$$

Since 100% of efficiency is required:

$$Po = Pin = 360W$$

The average inductor currents therefore are determined from the power and voltage specifications:

$$\mathbf{I_{L1}} = \frac{Ps}{Vs} = \frac{360W}{48V} = \mathbf{7.5A}$$

$$\mathbf{I_{L2}} = -\frac{Po}{Vo} = \frac{360W}{12V} = \mathbf{30A}$$

The 10% limit in changes in inductor currents requires: [Equation (4)&(5) applied]

$$\Delta i_L = \frac{VsD}{Lf}$$

$$\mathbf{\Delta i_{L1}} = 30A*10\% = \mathbf{3A}$$

$$\mathbf{\Delta i_{L2}} = 7.5A*10\% = \mathbf{0.75A}$$

With 50kHz switching frequency we can find the magnitudes of the inductances:

$$\mathbf{L_1} \geq \frac{VsD}{f\Delta i_{L1}} = \frac{48*0.33}{100k*0.75} = \mathbf{211.2uH}$$

$$\mathbf{L_2} \geq \frac{VsD}{f\Delta i_{L2}} = \frac{48*0.33}{100k*3} = \mathbf{52.8uH}$$

The requirement of the output ripple is no more than 1%, therefore, the value of output capacitance can be estimated: [Equation (2) applied]

$$\frac{\Delta Vo}{Vo} = \frac{1-D}{8L_2C_2f^2} = 1\%$$

$$C_2 \geq \frac{1-D}{1\%*8L_2*f^2} = \frac{1-0.33}{0.01*8*52.8\mu*(100k)^2}$$

$$\mathbf{C_2 \geq 15.86\mu F}$$

The average voltage across input capacitor C_1 is:

$$Vs\text{-}Vo = 48\text{-}(-12) = \mathbf{60V}$$

So the maximum change in v_{C1} (to match the 5% requirement) is:

$$\Delta v_{C1} = 60V*(0.05) = \mathbf{3V}$$

The equivalent load resistance is:

$$\mathbf{R} = \frac{Vo^2}{Po} = \frac{12^2}{360} = \mathbf{0.4\Omega}$$

With the 5% of voltage ripple variances, C1 can be calculated: [Equation (3) applied]

$$\Delta v_{C_1} \cong \frac{VoD}{RC_1f}$$

$$C_1 \geq \frac{VoD}{R\Delta v_{C_1}f} = \frac{12*0.33}{0.4*3*100k}$$

$$\mathbf{C_1 \geq 33\mu F}$$

Now, lets calculate the minimum inductance required to ensure continuous current flowing. [Equation (6) applied]

$$L_{1,\min} \geq \frac{(1-D)^2F}{2Df} = \frac{(1-0.33)^2*0.4}{2*0.33*100k}$$

$$\mathbf{L_{1,min} \geq 2.72\mu H}$$

$$L_{2,\min} \geq \frac{(1-D)F}{2f} = \frac{(1-0.33)*0.4}{2*100k}$$

$$\mathbf{L_{2,min} \geq 1.34\mu H}$$

APPENDIX B2

Simulation Waveform of the Cuk Converter

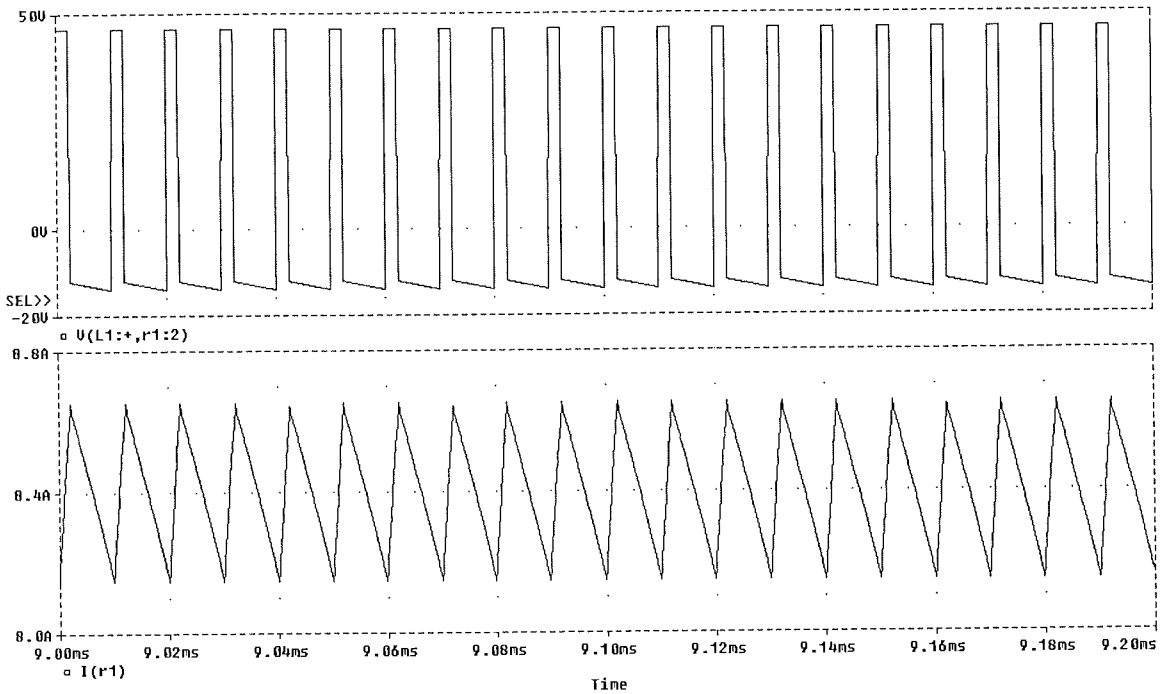


Figure C. Voltage (Top) and Current (Bottom) Waveform of the Inductor L1

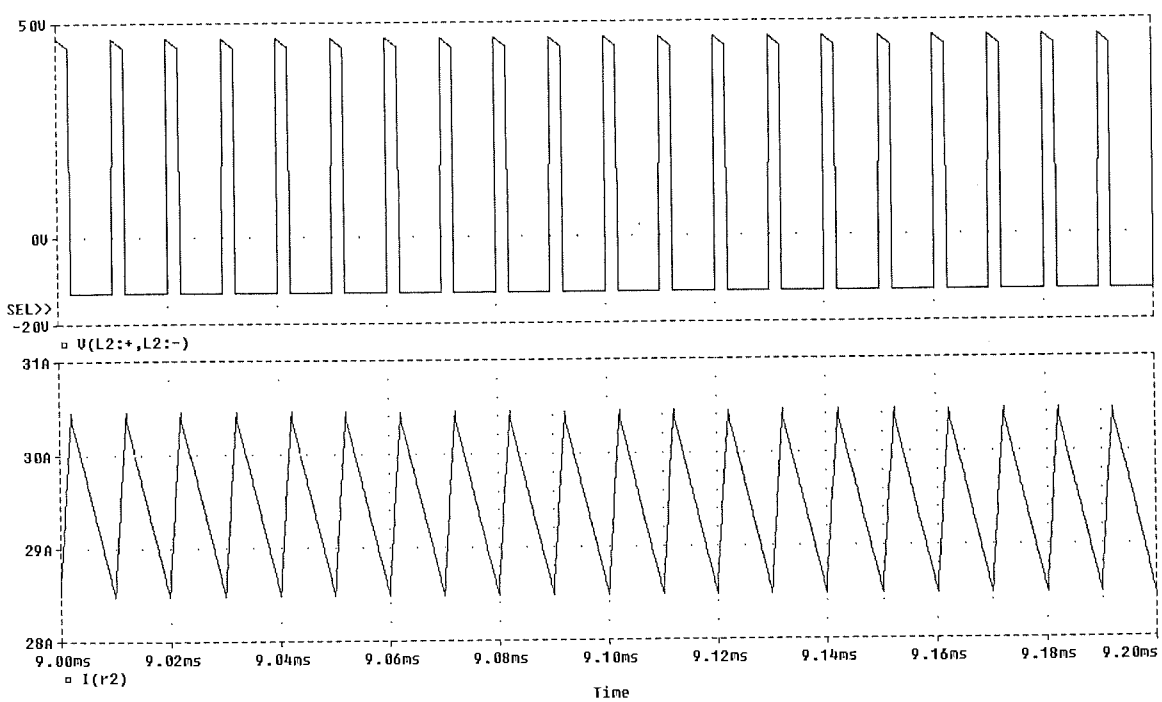


Figure D. Voltage (Top) and Current (Bottom) Waveform of the Inductor L2

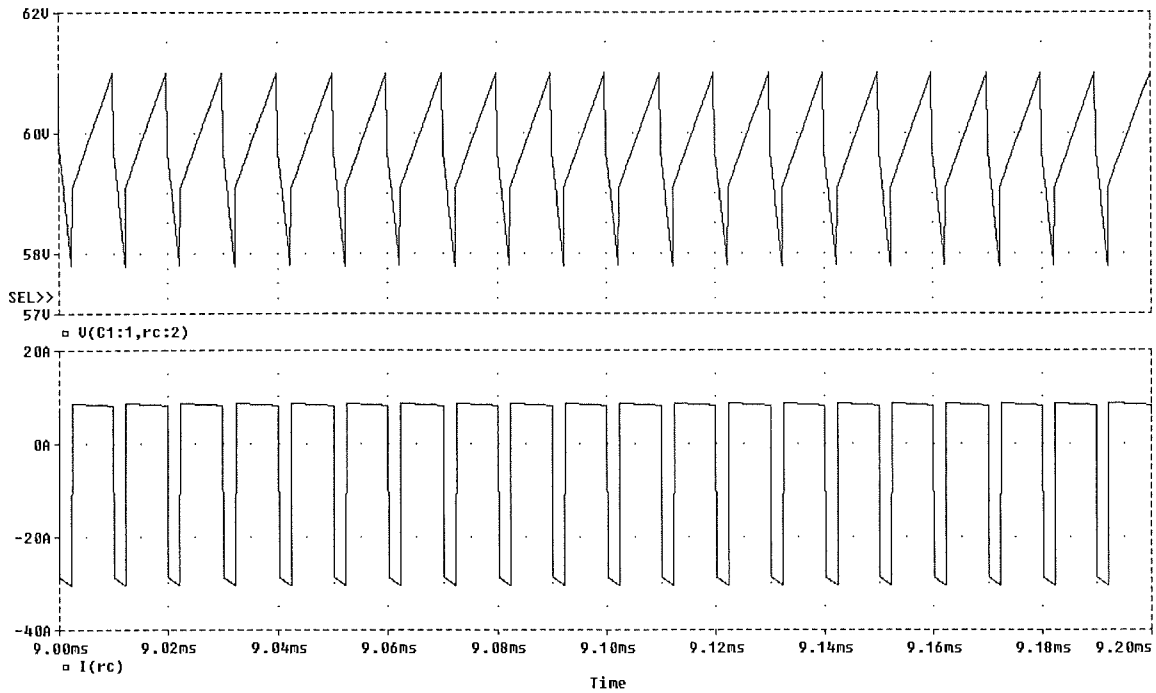


Figure E. Voltage (Top) and Current (Bottom) Waveform of the Capacitor C1

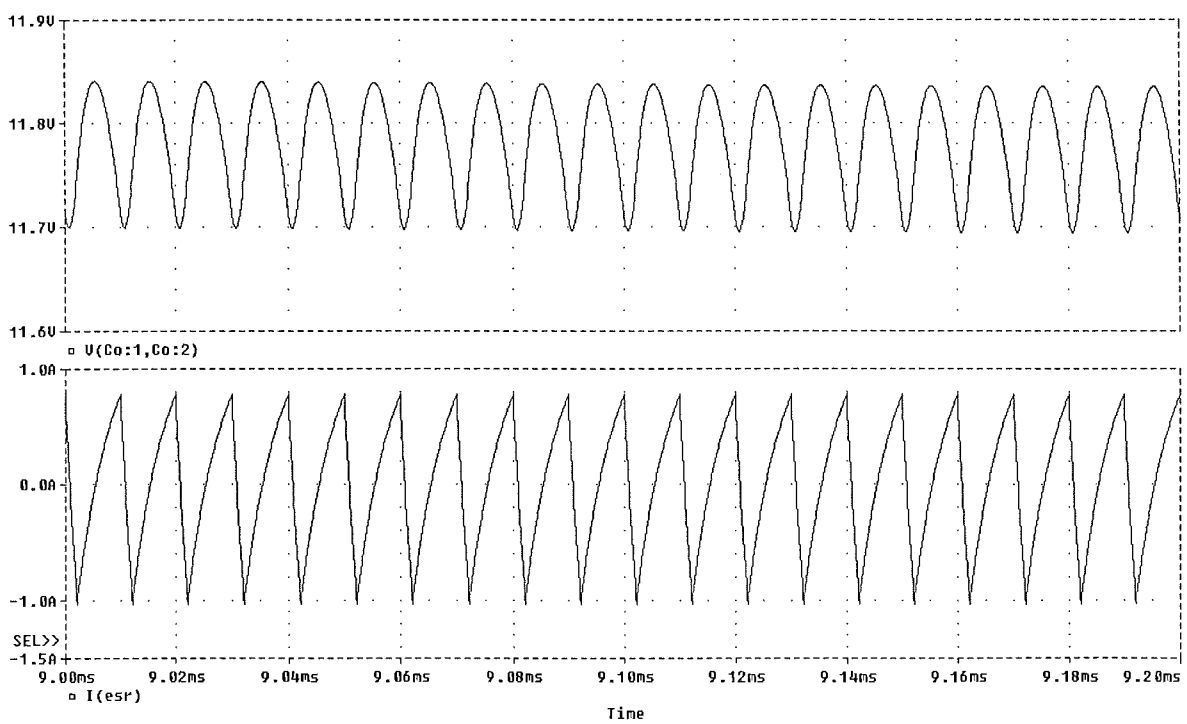


Figure F. Voltage (Top) and Current (Bottom) Waveform of the Capacitor C2

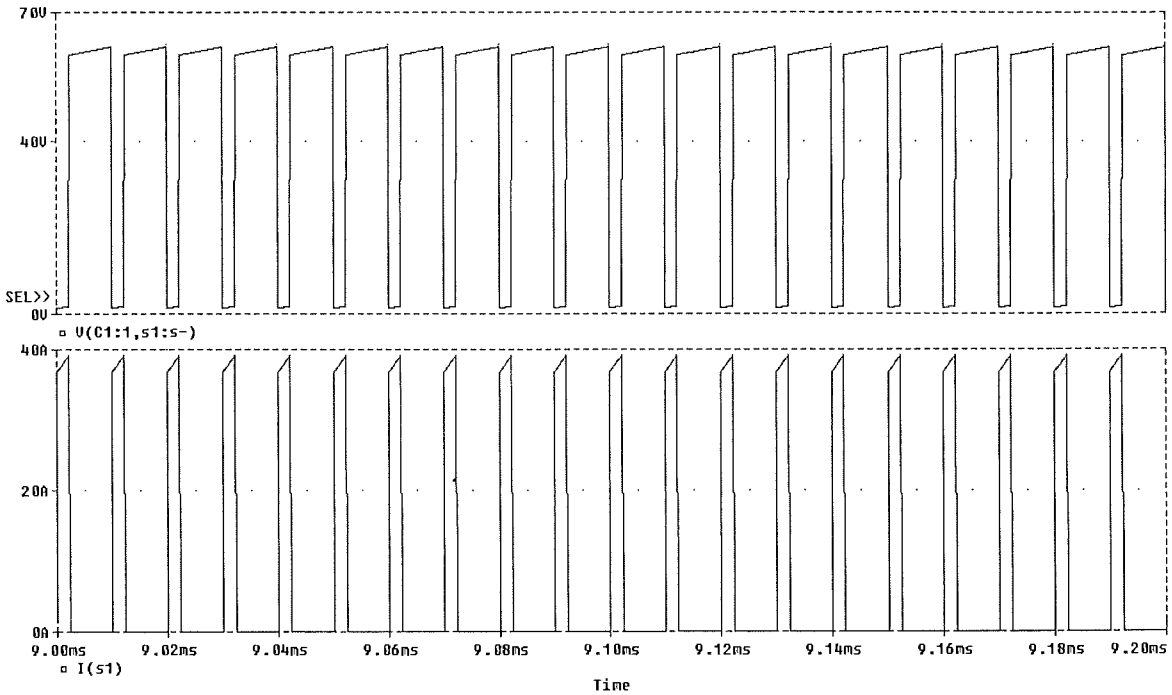


Figure G. Voltage (Top) and Current (Bottom) Waveform of the MOSFET S1

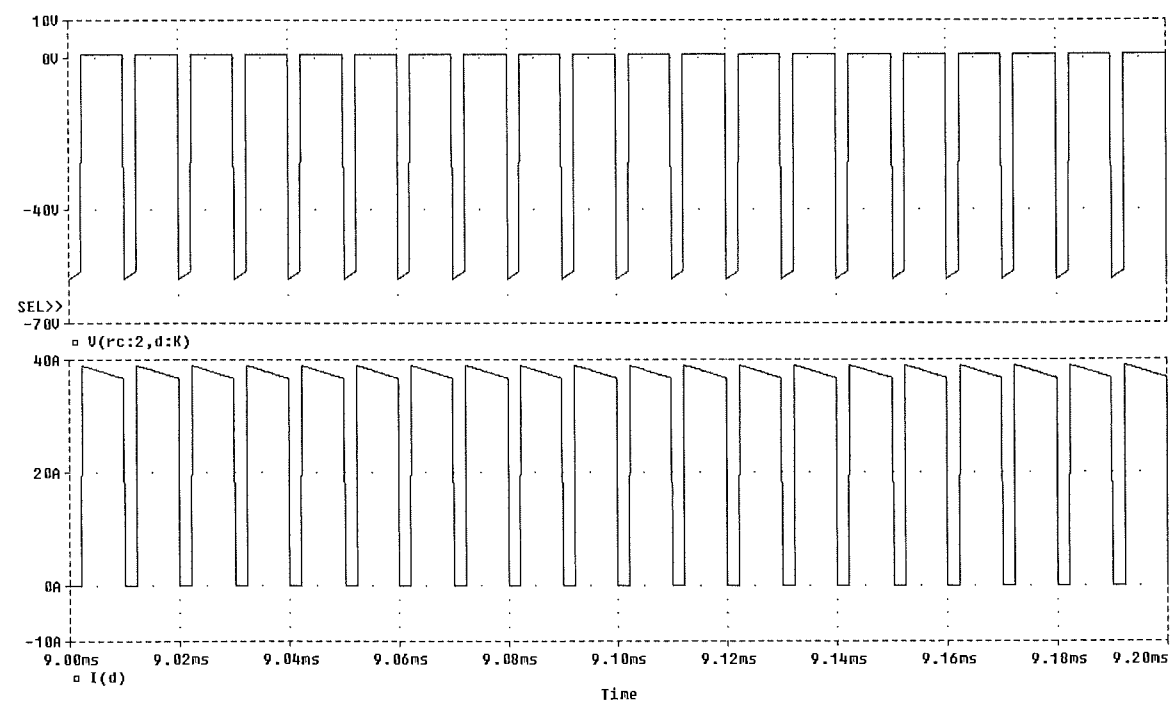


Figure H. Voltage (Top) and Current (Bottom) Waveform of the Diode D1

APPENDIX C1

Schematic of the CAN Control Node Board

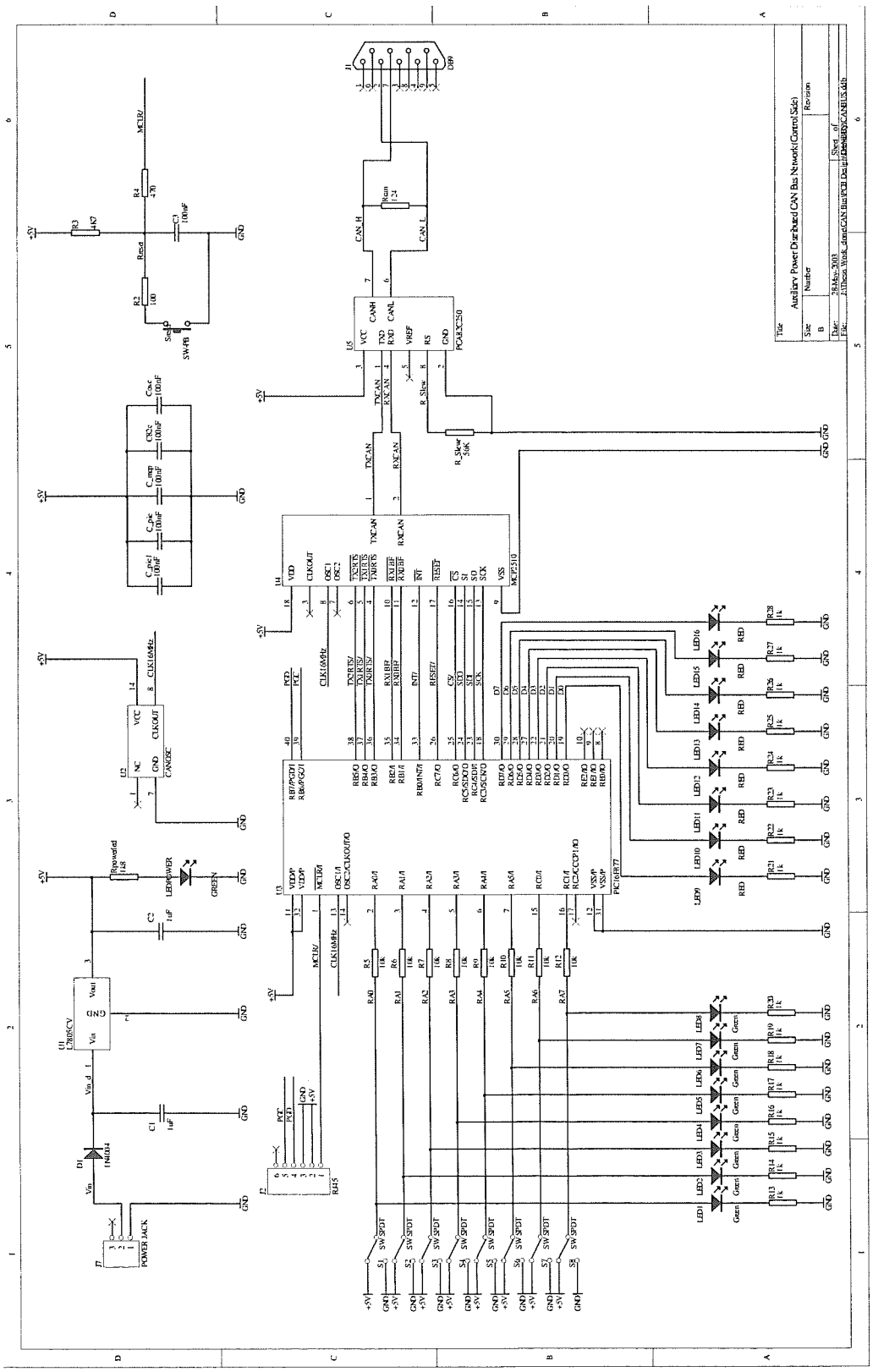


Figure I. Schematic circuitry of the CAN Control Node board

APPENDIX C2

Schematic of the CAN Load Node Board

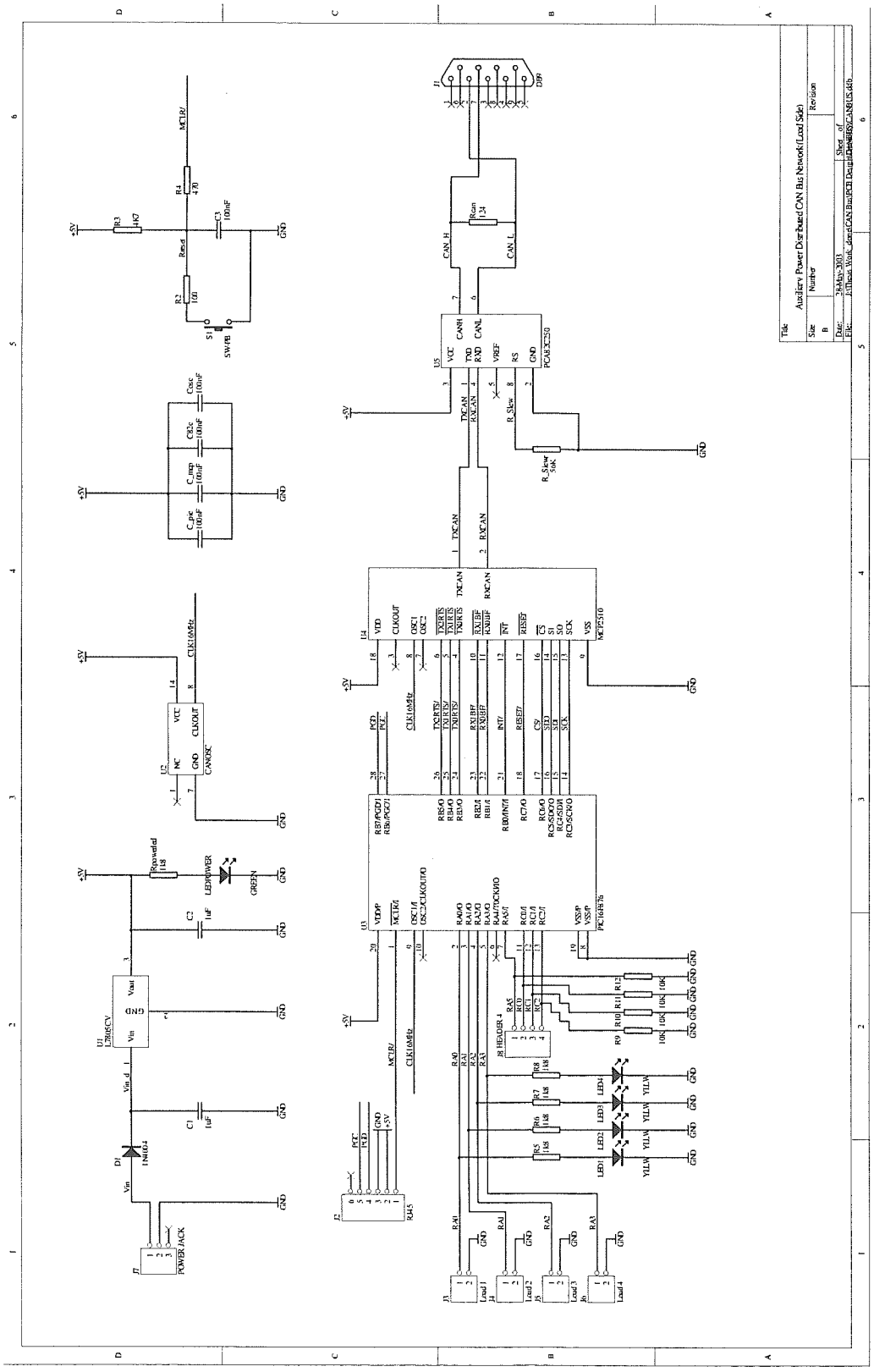


Figure J. Schematic circuitry of the CAN Load Node board

APPENDIX C3

Schematic of the Load Fault-Detector Board

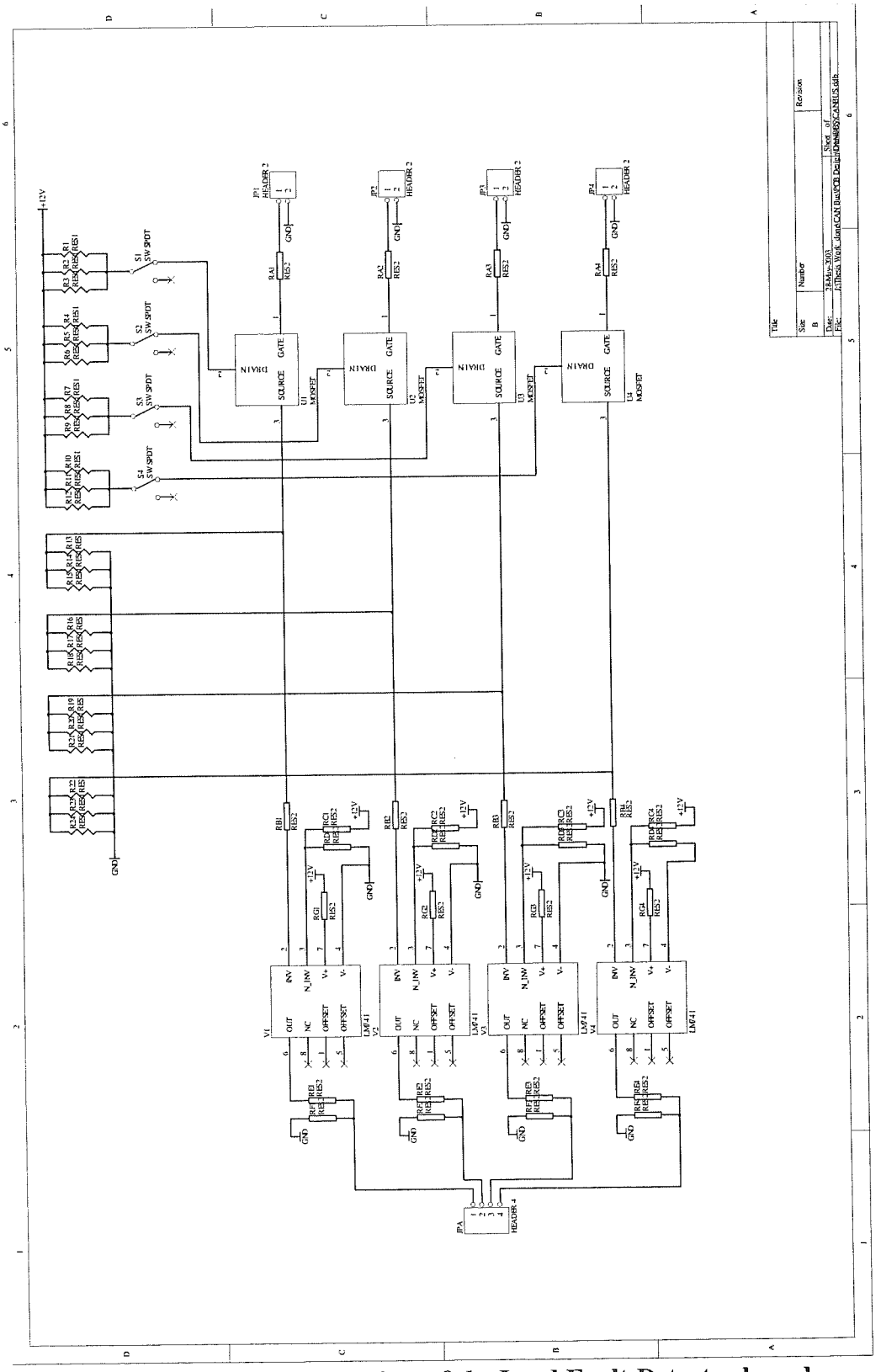


Figure K. Schematic circuitry of the Load Fault-Detector board

APPENDIX C4A

Software for the CAN Control Node

```
#include <pic.h>
#include "mcp.h"

// 125kBit/s @ 4MHZ Fosz
#define PRSEG   2
#define PHSEG1  7
#define PHSEG2  6
#define BRP     0
#define SJW     2

#define NODE    0
#define RXB0D 0x66
#define RXB1D 0x76

static MCP mcp;

static void init(){
  unsigned int delay;
  unsigned char i;

  ADCON1 = 0b110;
  PORTA = 0;
  TRISA = 0xFF;

  TRISC = 0b00010011;
  RC7 = 0; RC7 = 1;

  TRISD = 0;

  // Startup sequence
  PORTD = 1;
  for (i = 0; i < 15; i++){
    for (delay = 0; delay < 24000; delay++) continue;
    if (i < 7) PORTD <<= 1; else PORTD >>= 1;
   }

  //Init SPI, reset MCP2510
  mcp_init();
}

static unsigned char receiveCAN(MCP *mcp){
  mcp->operation = read;
  mcp->bufferLength = 1;

  mcp_rw(mcp);

  return mcp->buffer[0];
}

void main(){
  unsigned char switchIn, oldSwitchIn;
```

```
init();

//Write the CNF3 to 1 registers -> Bit time
mcp_setBitTimeHelper(&mcp, PRSEG, PHSEG1, PHSEG2, BRP, SJW);
mcp_rw(&mcp);

//Write the SID Acceptance Filter
mcp_setIDHelper(&mcp, RXF0ID, 1, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF1ID, 1, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF2ID, 2, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF3ID, 2, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF4ID, 2, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF5ID, 2, 0);
mcp_rw(&mcp);

//Write Identifier Mask Bits
mcp_setIDHelper(&mcp, RXM0ID, 0x7FF, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXM1ID, 0x7FF, 0);
mcp_rw(&mcp);

//No Rollover
mcp_setRegister(RXB0CTRL, 0b00100000);
mcp_setRegister(RXB1CTRL, 0b00100000);

//Write the SID register
mcp_setTXBnID(0,NODE,0);

mcp_setMode(normal);

while(1) {

  switchIn = PORTA | RC0 << 6 | RC1 << 7;

  if (switchIn != oldSwitchIn) {
    oldSwitchIn = switchIn;
    while(mcp_setTXBnD(0, &switchIn, 0, 1)) continue;
    mcp_setTXBnCTRL(0, 0, 1);
  }

  if (mcp_getStatus() & 0b11){
    if (mcp_getStatus() & MCP_RX1IF) {
      mcp.registerAddress = RXB1D;
      PORTD = receiveCAN(&mcp) | (PORTD & 0x0F);
      mcp_setBitModify(CANINTF, MCP_2C_RX1IF, 0);
    } else {
      mcp.registerAddress = RXB0D;
      PORTD = receiveCAN(&mcp) | (PORTD & 0xF0);
      mcp_setBitModify(CANINTF, MCP_2C_RX0IF, 0);
    }
  }
 }
}
```

APPENDIX C4B

Software for the CAN Load Node

```
#include <pic.h>
#include "mcp.h"

// 125kBit/s @ 4MHZ Fosz
#define PRSEG   2
#define PHSEG1  7
#define PHSEG2  6
#define BRP     0
#define SJW     2

#define NODE    2
#define RXB0D 0x66
#define RXB1D 0x76

static MCP mcp;

static void init(){
  unsigned int delay;
  unsigned char i;

  ADCON1 = 0b110;
  PORTA = 0;
  TRISA = 0b100000;

  TRISC = 0b00010111;
  RC7 = 0; RC7 = 1;

  // Startup sequence
  PORTA = 1;
  for (i = 0; i < 7; i++){
    for (delay = 0; delay < 48000; delay++) continue;
    if (i < 3) PORTA <<= 1; else PORTA >>= 1;
  }

  //Init SPI, reset MCP2510
  mcp_init();
}

static unsigned char receiveCAN(MCP *mcp){
  mcp->operation = read;
  mcp->bufferLength = 1;

  mcp_rw(mcp);

  return (NODE > 1) ? (mcp->buffer[0]>>4) : mcp->buffer[0];
}

void main(){
  unsigned char detector, oldDetector;

  init();
```

```
//Write the CNF3 to 1 registers -> Bit time
mcp_setBitTimeHelper(&mcp, PRSEG, PHSEG1, PHSEG2, BRP, SJW);
mcp_rw(&mcp);
//Write the SID Acceptance Filter
mcp_setIDHelper(&mcp, RXF0ID, 0, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF1ID, 0, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF2ID, 0, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF3ID, 0, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF4ID, 0, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXF5ID, 0, 0);
mcp_rw(&mcp);

//Write Identifier Mask Bits
mcp_setIDHelper(&mcp, RXM0ID, 0x7FF, 0);
mcp_rw(&mcp);
mcp_setIDHelper(&mcp, RXM1ID, 0x7FF, 0);
mcp_rw(&mcp);

//Set Rollover BIT RXB0 -> RXB1
mcp_setRegister(RXB0CTRL, 0b00100100);
mcp_setRegister(RXB1CTRL, 0b00100000);

//Write the SID register
mcp_setTXBnID(0,NODE,0);

mcp_setMode(normal);


while(1) {

  detector = RA5 | (PORTC & 0b111) << 1;
  if (NODE > 1) detector <<= 4;

  if (detector != oldDetector) {
    oldDetector = detector;
    while(mcp_setTXBnD(0, &detector, 0, 1)) continue;
    mcp_setTXBnCTRL(0, 0, 1);
  }

  if (mcp_getStatus() & 0b11){
    if (mcp_getStatus() & MCP_RX1IF) {
      mcp.registerAddress = RXB1D;
      PORTA = receiveCAN(&mcp);
      mcp_setBitModify(CANINTF, MCP_2C_RX1IF, 0);
    } else {
      mcp.registerAddress = RXB0D;
      PORTA = receiveCAN(&mcp);
      mcp_setBitModify(CANINTF, MCP_2C_RX0IF, 0);
    }
  }
 }
}
```

APPENDIX C5

Software Function Codes

```
/* Author: Philipp Hof
   Modified: William Chen
   READ: MCP2510 Rev. AA Silicon Errata Sheet
   The MCP2510 chip has many errors.
*/

#include <pic.h>
#include "mcp.h"

#define SPI_RXTX(tx)   SSPBUF = tx; \
                while(!STAT_BF) continue

void mcp_init(){
  unsigned char i;

  MCP_CS = 1;
  MCP_CS_TRIS = 0;
  TRISC5 = 0;
  TRISC3 = 0;
  STAT_CKE = 1;
  STAT_SMP = 0;
  SSPCON = 0b00100000;

  MCP_CS = 0;
  SPI_RXTX(reset);
  MCP_CS = 1;
  for (i = 0; i < 128; i++) continue; //wait 128 Tosc after reset
}


/* ************************** GENERIC FUNCTIONS ************************** */

void mcp_rw(MCP *mcp){
  unsigned char i = 0;

  MCP_CS = 0;
  SPI_RXTX(mcp->operation);

  switch (mcp->operation){
   case  read:
   case write: SPI_RXTX(mcp->registerAddress);
           for (i = 0; i < mcp->bufferLength; i++){
             SPI_RXTX(mcp->buffer[i]);
             if (mcp->operation == read) mcp->buffer[i] = SSPBUF;
           }
           break;
   case reset: for (i = 0; i < 128; i++) continue; //wait 128 Tosc after reset
           break;
   default   : break;
  }
  MCP_CS = 1;
}
```

```
/* ************************** CONTROL FUNCTIONS ************************** */

void mcp_setRegister(unsigned char address, unsigned char value){
  MCP_CS = 0;
  SPI_RXTX(write);
  SPI_RXTX(address);
  SPI_RXTX(value);
  MCP_CS = 1;
}

unsigned char mcp_getRegister(unsigned char address){
  MCP_CS = 0;
  SPI_RXTX(read);
  SPI_RXTX(address);
  SPI_RXTX(0);
  MCP_CS = 1;

  return SSPBUF;
}

MCP_MODE mcp_getMode(void){
  MCP_CS = 0;
  SPI_RXTX(read);
  SPI_RXTX(CANSTAT);
  SPI_RXTX(0);
  MCP_CS = 1;
  return SSPBUF >> 5;
}

void mcp_setMode(MCP_MODE mode){
  MCP_CS = 0;
  SPI_RXTX(write);
  SPI_RXTX(CANCTRL);
  SPI_RXTX((mode << 5) & 0b11100000);
  MCP_CS = 1;
}

void mcp_setRTSnBITS(unsigned char n){
  MCP_CS = 0;
  SPI_RXTX(0x80 | (n & 0b111));
  MCP_CS = 1;
}

unsigned char mcp_getStatus(){
  MCP_CS = 0;
  SPI_RXTX(0b10100000);
  SPI_RXTX(0);
  MCP_CS = 1;

  return SSPBUF;
}

void mcp_setBitModify(unsigned char address, unsigned char mask, unsigned char data){
  MCP_CS = 0;
  SPI_RXTX(0b101);
  SPI_RXTX(address);
  SPI_RXTX(mask);
  SPI_RXTX(data);
  MCP_CS = 1;
}
```

```
/* ************************** TRANSMIT FUNCTIONS ************************** */

void mcp_setTXBnCTRL(unsigned char n, unsigned char priority, unsigned char TXREQ){
  switch (n){
    case 0: n = TXB0CTRL; break;
    case 1: n = TXB1CTRL; break;
    case 2: n = TXB2CTRL; break;
    default: return;
  }

  if (priority > 3) priority = 3;

  MCP_CS = 0;
  SPI_RXTX(write);
  SPI_RXTX(n);
  SPI_RXTX(priority | (TXREQ & 1) << 3);
  MCP_CS = 1;
}

unsigned int mcp_setTXBnID(unsigned char n, unsigned long ID, unsigned char EXIDE){

  unsigned char EID7, EID15, EID23;
  unsigned char checkTXREQ = mcp_getStatus();

  switch (n){
    case 0: if (checkTXREQ & MCP_TXREQ0) return checkTXREQ;
          n = TXB0ID; break;
    case 1: if (checkTXREQ & MCP_TXREQ1) return checkTXREQ;
          n = TXB1ID; break;
    case 2: if (checkTXREQ & MCP_TXREQ2) return checkTXREQ;
          n = TXB2ID; break;
    default: return MCP_INVALID;
  }

  MCP_CS = 0;
  SPI_RXTX(write);
  SPI_RXTX(n);

  if (EXIDE){
    EID7 = ID;
    ID >>= 8;
    EID15 = ID;
    ID >>= 8;
    EID23 = ID & 0b11 | (ID & 0b11100) << 3 | 0b1000;
    SPI_RXTX(ID >> 5);
    SPI_RXTX(EID23);
    SPI_RXTX(EID15);
    SPI_RXTX(EID7);
  } else {
    SPI_RXTX(ID >> 3);
    SPI_RXTX(ID << 5);
  }

  MCP_CS = 1;

  return MCP_OKAY;
}
```

```
unsigned int mcp_setTXBnD(unsigned char n, const unsigned char *data, unsigned char RTR, unsigned char
length){

  unsigned char checkTXREQ = mcp_getStatus();

  switch (n){
    case 0: if (checkTXREQ & MCP_TXREQ0) return checkTXREQ;
          n = TXB0D; break;
    case 1: if (checkTXREQ & MCP_TXREQ1) return checkTXREQ;
          n = TXB1D; break;
    case 2: if (checkTXREQ & MCP_TXREQ2) return checkTXREQ;
          n = TXB2D; break;
    default: return MCP_INVALID;
  }

  if (length > 8) return MCP_INVALID;
  MCP_CS = 0;
  SPI_RXTX(write);
  SPI_RXTX(n);
  SPI_RXTX(length | (RTR & 1) << 6);

  while(length-- > 0) SPI_RXTX(*data++);
  MCP_CS = 1;

  return MCP_OKAY;
}


/* ************************* HELPER FUNCTIONS ************************* */

void mcp_setBitTimeHelper(MCP *mcp, unsigned char PRSEG, unsigned char PHSEG1, unsigned char
PHSEG2, unsigned char BRP, unsigned char SJW){
  mcp->operation = write;
  mcp->registerAddress = CNF3;
  mcp->buffer[0] = PHSEG2-1;
  mcp->buffer[1] = PRSEG-1 | (PHSEG1 - 1) << 3 | 0x80;
  mcp->buffer[2] = BRP | (SJW-1) << 6;
  mcp->bufferLength = 3;
}

void mcp_setIDHelper(MCP *mcp, unsigned char address, unsigned long ID, unsigned char EXIDE){
  mcp->operation = write;
  mcp->registerAddress = address;

  if (EXIDE){
    mcp->buffer[3] = ID;
    ID >>= 8;
    mcp->buffer[2] = ID;
    ID >>= 8;
    mcp->buffer[1] = ID & 0b11 | (ID & 0b11100) << 3 | 0b1000;
    mcp->buffer[0] = ID >> 5;
    mcp->bufferLength = 4;
  } else {
    mcp->buffer[0] = ID >> 3;
    mcp->buffer[1] = ID << 5;
    mcp->bufferLength = 2;
  }
}
```

```
void mcp_setTXBnDHelper(MCP *mcp, unsigned char n, const unsigned char *data, unsigned char RTR,
unsigned char length){

  switch (n){
    case 0: n = TXB0D; break;
    case 1: n = TXB1D; break;
    case 2: n = TXB2D; break;
    default: return;
  }

  if (length > 8) return;

  mcp->operation = write;
  mcp->registerAddress = n;
  mcp->bufferLength = length + 1;

  mcp->buffer[0] = length | (RTR & 1) << 6;

  for (length = 0; length < mcp->bufferLength-1; length++){
    mcp->buffer[length+1] = data[length];
  }
}


/* ************************** RECEIVE FUNCTIONS ************************** */

unsigned char mcp_getRXBnDataLength(unsigned char n){

  switch (n){
    case 0: n = RXB0DLC; break;
    case 1: n = RXB1DLC; break;
    default: return 0;
  }

  MCP_CS = 0;
  SPI_RXTX(read);
  SPI_RXTX(n);
  SPI_RXTX(0);
  MCP_CS = 1;

  return SSPBUF & 0b1111;
}

void mcp_getRXBnID(MCP_RX *mcpRx, unsigned char n){

  switch (n){
    case 0: n = RXB0ID; break;
    case 1: n = RXB1ID; break;
    default: return;
  }

  MCP_CS = 0;

  SPI_RXTX(read);
  SPI_RXTX(n);

  SPI_RXTX(0);
  mcpRx->ID = SSPBUF;

  SPI_RXTX(0);
```

```
  mcpRx->remTXReq = none;
  if (SSPBUF & 0b1000){
    mcpRx->frame = extended;
    mcpRx->ID = mcpRx->ID << 5;
    mcpRx->ID = mcpRx->ID | (SSPBUF >> 3) & 0b11100 | SSPBUF & 0b11;

    SPI_RXTX(0);
    mcpRx->ID = (mcpRx->ID<<8) | SSPBUF;

    SPI_RXTX(0);
    mcpRx->ID = (mcpRx->ID<<8) | SSPBUF;

  } else {
    mcpRx->frame = standard;
    if (SSPBUF & 0b10000) mcpRx->remTXReq = standard;

    mcpRx->ID = (mcpRx->ID << 3) | (SSPBUF >> 5);
    SPI_RXTX(0);
    SPI_RXTX(0);
  }

  SPI_RXTX(0);
  mcpRx->dataLength = SSPBUF & 0b1111;

  if ((mcpRx->frame == extended) && (SSPBUF & 0b01000000)) mcpRx->remTXReq = extended;

  MCP_CS = 1;
}

void mcp_getRXBn(MCP *mcp, MCP_RX *mcpRx, unsigned char n){

  if (n > 1) return;
  mcp_getRXBnID(mcpRx, n);

  mcp->operation = read;
  mcp->bufferLength = mcpRx->dataLength;

  switch (n){
    case 0: mcp->registerAddress = RXB0DLC+1; break;
    case 1: mcp->registerAddress = RXB1DLC+1; break;
    default: return;
  }

  mcp_rw(mcp);
}
```