

UNIVERSITY OF SOUTHERN QUEENSLAND  
Faculty of Engineering and Surveying

**A Four Quadrant Adjustable Speed Drive  
For Series Wound DC Motors**

A dissertation submitted by

**Dean Thompson**

in fulfilment of the requirements of

**ENG4112 – Research Project**

towards the degree of

**Bachelor of Engineering (Electrical & Electronic)**

Submitted: October 2005

# Abstract

The aim of this research project is to design and test a motor speed controller for use in medium sized electric vehicles such as golf carts and forklift trucks. The adjustable speed drive provides control of a series motor in four modes; forward motoring, forward regeneration (i.e. braking), reverse motoring and reverse regeneration. A low-cost design approach was taken, with a microcontroller forming the heart of the design. Most of the drive's features are implemented in software. Power stage design was also considered, and a suitable power electronic circuit to interface to the control system was designed.

Motor control is a complex topic due to the wide range of issues involved. The following report discusses these issues, including motor theory, digital and analog electronic design, high power DC-DC converters, thermal design, EMI and EMC issues, and software design.

The completion of this project has demonstrated the advantages of using a microcontroller as the basis for the design and proved the feasibility of using a highly integrated approach to the circuit design. Testing of the control system showed that its performance is suitable for the application and highlighted the benefits of using software to incorporate many useful features in the design with a low parts count and minimal cost.

UNIVERSITY OF SOUTHERN QUEENSLAND  
Faculty of Engineering and Surveying

**ENG4111/2 *Research Project***

**Limitations of Use**

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Prof G Baker  
Dean  
Faculty of Engineering and Surveying

# Certification

I Certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.



23/10/2005

**Dean Thompson**

**Student Number: Q1222857**

# Table of Contents

<i>Section</i>	<i>Page</i>
List of Figures and Tables .....	iii
Acknowledgements.....	v
Glossary .....	vi
1.0 Introduction .....	1
Section 1.1 – Background.....	1
Section 1.2 – Aims and Objectives.....	3
Section 1.3 – Outline of Dissertation.....	6
2.0 Principles of Operation.....	7
Section 2.1 – Series Wound DC Motor Model.....	7
Section 2.2 – DC-DC Converter Operation.....	13
Section 2.3 – The Selected Power Electronic Configuration .....	17
Section 2.4 – Gate Drive .....	20
Section 2.5 – Software Requirements .....	24
Section 2.6 – Protection.....	25
3.0 Prototyping and Testing of the Design Concepts.....	27
Section 3.1 – Microcontroller Modules .....	27
Section 3.2 – Analog Circuits .....	36
Section 3.3 – Gate Drive .....	42
Section 3.4 – Support Circuits.....	45
4.0 Control System Implementation and Testing.....	47
Section 4.1 – Overall Description.....	47
Section 4.2 – Microcontroller Implementation.....	50
Section 4.3 – EMC and EMI Considerations.....	53
Section 4.4 – Software.....	56
Section 4.5 – Overall Circuit Operation.....	61
Section 4.6 – Control System Testing.....	63
5.0 Power Electronic Design.....	70
Section 5.1 – Selection of Power Switching Devices .....	70
Section 5.2 – Thermal Design .....	76
Section 5.3 – Layout, EMI and EMC Issues .....	89
6.0 Conclusion.....	92
Section 6.1 – Achievement of Objectives .....	92
Section 6.2 – Further Work .....	94
List of References .....	95
Bibliography .....	96
Appendix A – Project Specification .....	97
Appendix B – PIC16F877A Data Sheet Extracts .....	99

Appendix C – PIC Programmer Schematic Diagram.....	103
Appendix D – Power Stage and Control Circuit Diagrams.....	105
Appendix E – Microcontroller Software Code Listing.....	108
Appendix F – PC Software Code Listing.....	116

# List of Figures and Tables

<i>Number</i>	<i>Page</i>
Figure 1.1 – Simple System Block Diagram.....	4
Figure 2.1 – The Force on a Conductor (Hill 2003) .....	7
Figure 2.2 – Series and Shunt Type Configurations.....	8
Figure 2.3 – Motor Circuit Model.....	9
Figure 2.4 – Series Wound Motor Current-Torque Curve .....	11
Figure 2.5 – Series Wound DC Motor Torque-Speed Curve.....	12
Figure 2.6 – Buck Converter Circuit .....	13
Figure 2.7 – Boost Converter Circuit.....	14
Figure 2.8 – Continuous Mode Voltage and Current Waveform of a Buck Converter .....	15
Figure 2.9 – H Bridge Configuration .....	17
Figure 2.10 – Half Bridge with Reversing Contactors .....	18
Figure 2.11 – Four Quadrants of Operation .....	19
Figure 2.12 – Simplified MOSFET Model and Gate Driver (Davis 2003).....	21
Figure 2.13 – Power MOSFET Gate Drive Characteristics (Davis 2003).....	22
Figure 3.1 – PIC Programmer .....	28
Figure 3.2 – Testing the LCD.....	29
Figure 3.3 – ADC Testing.....	32
Figure 3.4 – PWM Output (Microchip 2003).....	33
Figure 3.5 – PWM Module Block Diagram (Microchip 2003).....	33
Figure 3.6 – PWM Test Circuit .....	34
Figure 3.7 – Microcontroller’s PWM Output Signal .....	34
Figure 3.8 – MAX232 RS232 Driver Circuit .....	35
Figure 3.9 – Motor Current Sense Amplifier.....	37
Figure 3.10 – Surface Mount to DIP Adaptor PCB.....	38
Figure 3.11 – LM2917 Tachometer Circuit .....	40
Figure 3.12 – Accelerator and Decelerator Inputs .....	40
Figure 3.13 – Programmable Shunt Regulator Voltage Reference.....	41
Figure 3.14 – Half Bridge MOSFET Driver Circuit .....	42
Figure 3.15 – Half Bridge MOSFET Driver Test Waveforms .....	43
Figure 3.16 – Shoot Through Protection Logic .....	44
Figure 3.17 – Power Supply .....	45
Figure 4.1 – Functional Block Diagram.....	47

Figure 4.2 – Microcontroller Inputs and Outputs .....	50
Figure 4.3 – Poor Motor Control Design (Valentine 1998).....	52
Figure 4.4 – Improved Motor Control Design (Valentine 1998) .....	52
Figure 4.5 – Electrical Noise Problem Areas in Motor Drive Systems (Valentine 1998).....	53
Figure 4.6 – Screenshot of PC Configuration Software .....	59
Table 4.1 – Drive Parameters .....	60
Table 4.2 – Microcontroller Pin Assignment.....	61
Figure 4.7 – Testing of the Control System.....	63
Figure 4.8 – PWM Output During Normal Operation.....	65
Figure 4.9 – Current Limit Response Time .....	67
Figure 4.10 – Speed Limit Response Time .....	69
Table 5.1 – Power Device Attributes .....	71
Figure 5.1 – Basic Heat Conduction Model (Valentine 1998).....	78
Table 5.2 – Interface Material Properties .....	83
Table 5.3 – Heatsink Thermal Resistances (Valentine 1998).....	84
Figure 5.2 – Motor Controller Layout (Valentine 1998) .....	90
Figure 5.3 – Enclosure Layout .....	91



# Acknowledgments

This project could not have been completed without the help and support of many people. Especially, I would like to thank Dr Tony Ahfock for his continual input and assistance, which has been invaluable. The hardware design aspect of the project could not have been completed without the help of Mr Terry Byrne and Mr Brett Richards fulfilling my constant requests for parts, and fabricating the special SMD adaptor PCB.

I would also like to thank Mr Ian Bentley for allowing me to source excellent books from the University of Queensland library for an extended loan period. Of course, in addition, I thank all my friends and family for their ever present support and understanding.

# Glossary

**ADC.** Analog to Digital Converter

**Acquisition Time.** The time taken for an ADC's sample & hold capacitor to charge

**ASD.** Adjustable Speed Drive

**BJT.** Bipolar Junction Transistor

**Conversion Time.** The time taken for an ADC to convert its input voltage to a digital value

**EEPROM.** Electronically Erasable Programmable Read Only Memory

**EMC.** Electromagnetic Compatibility

**EMI.** Electromagnetic Interference

**EV.** Electric Vehicle

**ICSP.** In Circuit Serial Programming

**IGBT.** Insulated Gate Bipolar Transistor

**MOSFET.** Metal-Oxide Silicon Field Effect Transistor

**MOV.** Metal-Oxide Varistor

**PIC.** Programmable Integrated Circuit

**PWM.** Pulse Width Modulation

**RISC.** Reduced Instruction Set Computer

**SCR.** Silicon Controlled Rectifier

**SMPS.** Switch Mode Power Supply

**USART.** Universal Synchronous Asynchronous Receiver Transmitter

**WOT.** Wide Open Throttle

# Chapter 1

## 1.0 Introduction

This project encompasses the design, construction and testing of a four quadrant DC adjustable speed drive for an electric vehicle (EV) with a series-wound traction motor. The proposed drive system will consist of a power electronic circuit and its associated control logic, powered by an on-board 48V battery pack.

Four quadrant operation means the drive can operate the motor in four different modes by changing the polarity of voltage and direction of current. These four modes are: forward motoring (positive voltage and current), forward regeneration (positive voltage, negative current), reverse motoring (negative voltage and current) and reverse regeneration (negative voltage and positive current). The term regeneration (also known as regenerative braking) means operating the motor as a generator. This brakes the motor by converting its mechanical energy into electrical energy and sending it back to the batteries.

Protection systems must also be implemented and will include speed limiting and current limiting, as well as the required safety interlocks. The specific application is a recreational four-wheeled vehicle consisting of a standard go-kart chassis, however the design could be implemented on any EV with a series wound DC motor such as a forklift truck or golf cart.

## **Section 1.1 – Background**

### **1.1.1 Why a Series Wound Motor Was Chosen**

Series wound motors are commonly used as traction motors in many applications, as they offer high starting torque, are robust, have a simple design and are relatively low cost. Most of their applications are of an industrial nature, such as forklifts and conveyors, but are also common in road-going electric vehicles and railways (including trams and locomotives). DC series motors are an ideal choice for battery-operated equipment over AC motors, as they do not require the use of expensive inverter circuitry to convert the DC voltage to an AC voltage required by the motor. The major disadvantage of DC motors is that they require a commutator and brushes, which wear over time and thus require some maintenance.

### **1.1.2 Why Use Power Electronics**

There have been many methods of motor control invented over the past hundred years or so. They range from simple rheostat type controllers, which basically are a big variable resistor, to voltage-step contactor controls, to the modern electronic versions available today. Before the advent of power electronics, most methods of motor control were inefficient or did not provide smooth operation. Modern variable speed drives are very efficient and use transistor technology to control power to the motor smoothly and safely.

Variable speed drives are used wherever there is a need to vary the speed and torque of the motor. They are used mainly in industrial applications controlling machinery or in electric vehicles such as forklifts. Smaller versions can also be found in items like cordless drills and remote control cars.

### **1.1.3 Why Control Voltage with the Switch-Mode Method**

Voltage control using the switch-mode concept utilises solid state components, including high power transistors driven by PWM (pulse width modulation). The benefits of using this method for motor control are many, in particular;

- The ability to smoothly control voltage (and current) delivered to the motor, thus controlling its speed and torque.
- All operating modes can be realised, including forward, reverse and regenerative braking.
- High efficiency – losses can be minimised and an efficiency of 80% or higher is easily obtainable.
- Easily implemented into digital circuits, allowing high levels of controllability and large feature sets to be implemented in software.

## Section 1.2 – Aims and Objectives

The general aims of the project are set out in the project specification given in Appendix A. The main objective of this project is to design an economical drive that will allow four-quadrant operation of series wound motors in an EV application. While there are commercial series motor drives available from companies such as Curtis Instruments, these are generally only of the single or two quadrant types. Regeneration with series wound motors is uncommon due to the construction of the motor. Modern power electronics make regeneration with these types of motors more feasible. With the aid of the power electronic circuit used in this project, the effectiveness of regeneration with these motors could be investigated.

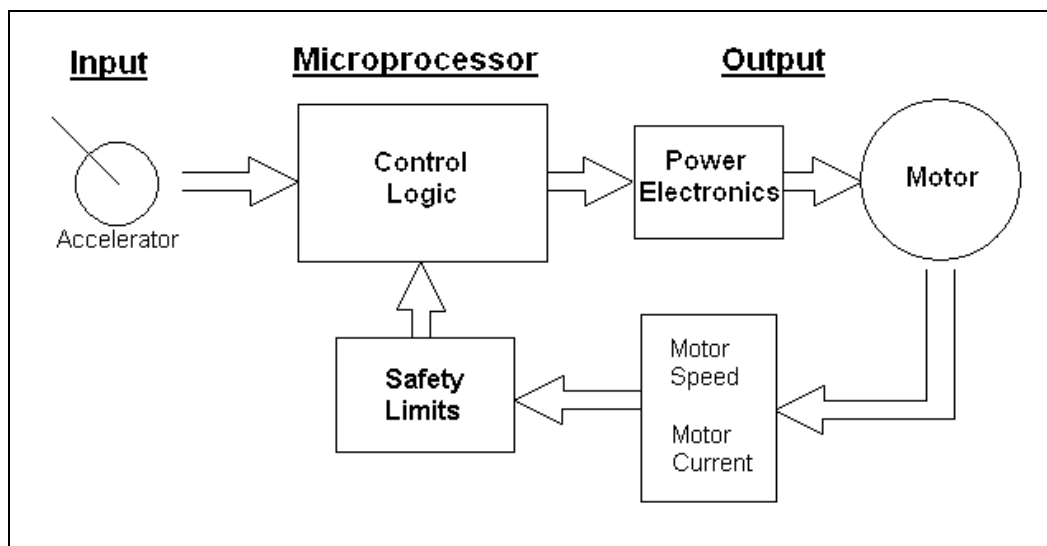
To achieve the objectives of the project with low cost and minimal complexity, the following parameters were chosen as the basis for the design:

- Half-Bridge design with MOSFET switches
- Mechanical switching of motor connection for direction reversal
- Microcontroller with integrated ADC, PWM and USART features for control logic
- Use of dedicated ICs for gate drive, current sensing, etc instead of discrete devices.

Taking this approach will allow for the construction of a full-featured design that can be constructed for a minimum cost. The use of a microcontroller allows most of the features to be implemented in software, thus reducing the number of parts required in the design.

Since the intended application of the design is an EV operated by a human, an open-loop control system for motor speed can be implemented (no speed feedback to the controller). This is mainly because vehicles where pedal position is proportional to vehicle speed don't feel intuitive to the operator. Just like the accelerator pedal in a car varies engine speed and power by controlling the fuel intake, the pedal on an EV control should not directly vary the speed of the vehicle, but vary the power applied. Of course, there needs to be some feedback to the controller to ensure that the motor is operating within safe limits; however this feedback is for limiting purposes only.

Figure 1.1 below illustrates the basic blocks of the drive and how they interact.



**Figure 1.1 – Simple System Block Diagram**

Considering the application of the drive is an EV, the following features were seen as useful and as many of these as possible should be implemented in the design:

- LCD display output showing useful information for the operator
- Limits and parameters fully adjustable via RS232 PC link
- Separate current limits for forward, reverse and regeneration
- Speed limits for forward and reverse
- High and low battery voltage cut outs
- Battery overcharge protection during regeneration
- Main contactor control with capacitor pre-charge circuit
- Operator interface via forward/reverse selector switch and drive and regen controlled by two separate pedals (accelerator and brake)

There are four main goals that are to be achieved by this project;

- Design
- Test
- Construct
- Document

These goals must be achieved in the given timeframe and hence a sequence of events and timeline were determined. This list gives a set of the basic events that needed to be carried out;

1. Select a power electronic topology
2. Select a control system
3. Research and select appropriate parts for design
4. Order required parts
5. Acquire the required development tools (programmers, software, etc)
6. Familiarisation with the microcontroller
7. Initial circuit design
8. Prototyping of initial design
9. Programming of software for microcontroller
10. Initial testing and fine tuning
11. Revision and further testing as required through design phase
12. Documentation should occur over the entire period

## Section 1.3 – Outline of Dissertation

The dissertation consists of six chapters, the following five chapters address the aims and objectives set out in this introductory chapter. A brief overview the content of each chapter is given below for the benefit of the reader.

- Chapter 1 Introduction and presentation of the aims and objectives of the project.
- Chapter 2 Discusses the background theory of DC machines and DC-DC converters. This chapter also covers the selection of the appropriate power electronic topology, gate drive considerations, and software and protection requirements.
- Chapter 3 This chapter covers the design and testing of the various circuit blocks that form the design. These include the microcontroller and how it is programmed, current sense circuit, gate drive circuit and protection logic, RS232 interface circuit, speed sensing circuit, the power supply, as well as other elements of the design.
- Chapter 4 Discusses the system as a whole, including an overall description, how the microcontroller forms the control system, EMI and EMC issues in the control circuit, system software and finally testing of the control system.
- Chapter 5 Focuses on the power electronic design, which covers the selection of appropriate power devices, thermal design, EMI and EMC issues, and the layout and physical construction of the power stage.
- Chapter 6 This final chapter gives conclusions with respect to the overall project, discussing the achievement of objectives and the potential for further work in a number of areas.



# Chapter 2

## 2.0 Principles of Operation

### Section 2.1 – Series Wound DC Motor Model

This section explains the basic construction and characteristics of series wound DC motors. An electric motor is a device which converts electrical energy to mechanical energy. The operating behaviour of a series DC motor can be understood by mathematical models, these models relate the inputs of current and voltage to the mechanical output of the motor, namely torque and speed.

#### 2.1.1 Motor Principles

All electric motors operate using the interaction of two magnetic fields. There is one produced by a stationary part of the motor (this is called the stator, or field), and another by the rotating part (the rotor, or armature). These magnetic fields are produced by either energised coils of wire (windings) or permanent magnets.

Basic physics tells us that a current carrying conductor will produce a magnetic field. If this conductor is subjected to another external magnetic field, there will be a force exerted on the conductor. This follows Fleming's Left Hand Rule; the concept is illustrated in figure 2.1 below. (Hill 2003)

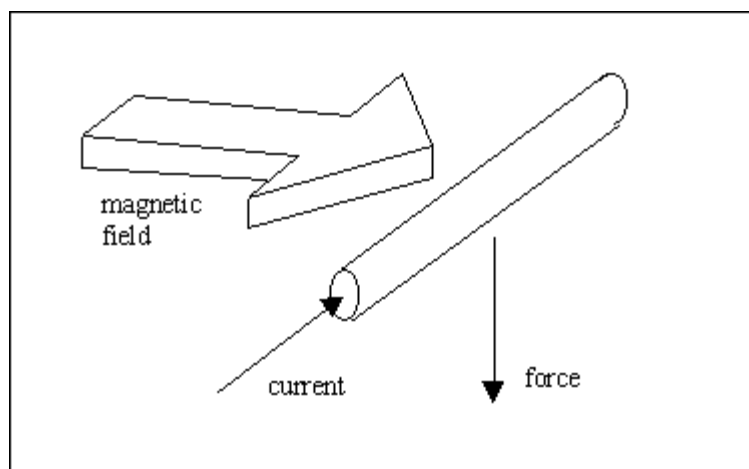


Figure 2.1 – The Force on a Conductor (Hill 2003)

A DC electric motor utilises this effect to produce rotational motion. The wire is bent into a loop, and placed in a magnetic field created by two magnets. In this case, both sides of the wire loop will have a force on them, trying to make the wire loop rotate. The current in the wire must reverse after one half rotation to keep the wire in motion. To do this, the current is applied to the loop through a commutator. This is basically a copper conductor with bars that alternate the connection of the wire loops. Current is applied to the commutator by a stationary block of graphite, called brushes, which rub against the commutator ring. Commutation is the process of changing the current direction in the coils.

In a practical motor many wire loops (called windings) are required at varying angles around a solid iron core to provide smooth rotation. Each wire loop has its own pair of commutator segments. This group of windings and its core is called the rotor or the armature.

The fixed magnets generating the field can be replaced by electromagnets. The electromagnets are supplied by the same power supply as the armature winding, either in series (series connected) or in parallel (shunt connected). If permanent magnets are used, the motor is said to be a permanent magnet (PM) motor. These stationary magnets (or electromagnets) form part of the stator.

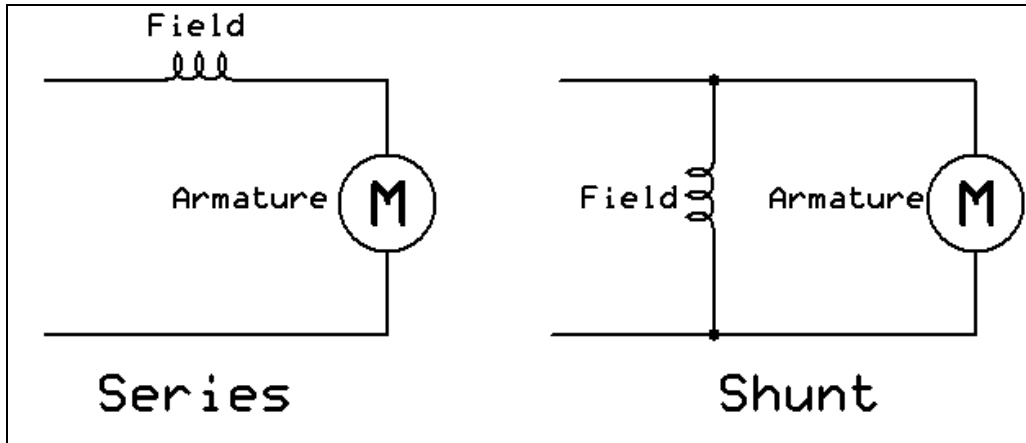


Figure 2.2 – Series and Shunt Type Configurations

It is possible to use more than just two fixed magnets (called poles). In some motors there may be four or more poles. There must always be an even number of poles, since there must be a north pole for every south pole.

A simple circuit model of a wound field motor can be drawn and is shown in figure 2.3.  $R_a$  is the resistance of the armature,  $R_f$  the resistance of the field and  $L_a$  and  $L_f$  are the inductance of the armature and field respectively.

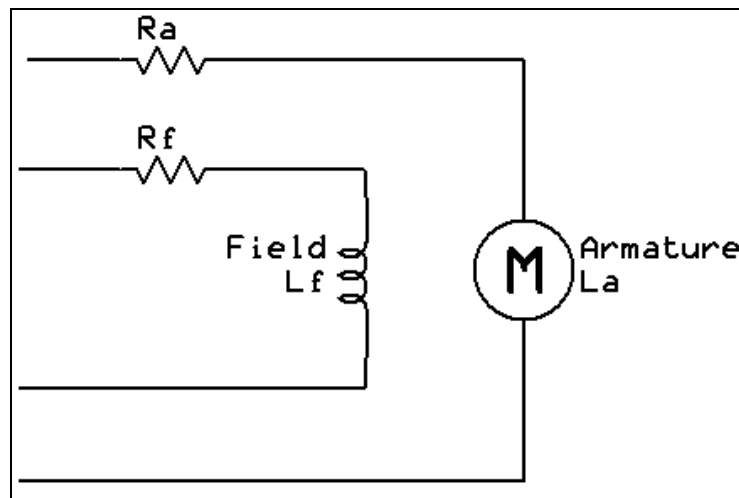


Figure 2.3 – Motor Circuit Model

### 2.1.2 Back EMF

A moving coil of wire will have a voltage induced in it when it is subjected to changes of magnetic flux. Therefore, as the magnetic forces cause the armature to rotate there is also a voltage produced across the windings in the armature, called the “back EMF”. When the motor is at steady state, this voltage closely matches the supply voltage applied to the motor. Faraday’s Law gives us the EMF induced in a coil, so an equation for the back EMF can be derived. Given an armature with a number ( $z$ ) of conductors in parallel paths ( $\alpha$ ) and a total flux on the armature surface produced by the number of poles ( $p\Phi$ ), the armature rotates at  $2\pi n$  radians per second. Each conductor will link with this flux  $n$  times per second. Applying Faraday’s Law, EMF produced around a single conductor is given by the flux linked per second. Therefore, the flux surrounding a conductor =  $p\Phi n$ . As stated previously, the armature has  $z$  conductors in  $\alpha$  parallel paths, thus the number of conductors in series is  $z \div \alpha$ . Thus the total EMF is given by;

$$E = \left( \frac{z}{\alpha} \right) p\Phi n$$

Combining the constant terms and using speed in rads/s:

$$E = \left( \frac{pz}{2\pi\alpha} \right) \Phi \omega$$

$$= K_e \Phi \omega$$

Where  $K_e = \left( \frac{pz}{2\pi\alpha} \right) = \text{EMF constant}$

By Ohm's Law, the current drawn by the motor can be calculated. Motor current will be given by:

$$I = \frac{\text{supply voltage} - \text{back EMF}}{\text{motor resistance}}$$

$$= \frac{V - (K_e \Phi \omega)}{R_t}$$

From this equation it can be seen that when the motor is at zero speed, the back EMF will be zero, and so the current taken by the motor will be very large. When the motor is spinning and the load is increased its speed will drop, so the back EMF drops, hence the current drawn by the motor will increase.

### 2.1.3 Torque

The torque produced by a permanent magnet or separately excited motor is given by the equation (assuming no losses):

$$T = k_t \Phi i_a$$

where  $k_t = k_e =$  the EMF constant for the motor, and  $i_a$  is the armature current, and  $\Phi$  is the field strength.

The field strength of a wound-field DC motor (such as a shunt or series motor) is given by:

$$\Phi = k_f i_f \quad \text{where } k_f \text{ is the field constant and } i_f \text{ is the field current.}$$

Due to the series connection of the field and armature winding, the series wound motor's field current is the same as the armature current, so  $i_a = i_f$ . Substituting this into the torque equation gives:

$$T = k_t \Phi i_a$$

$$= k_t k_f i_f i_a$$

$$i_a = i_f, \quad \therefore$$

$$T = k_t k_f i_a^2$$

Combining the constants results in the equation:

$$T = k i_a^2$$

Therefore, as the motor draws more current the field strength ( $\Phi$ ) increases. This causes the torque to be proportional to the square of the motor current. In effect, this means that doubling the current in a series wound motor quadruples the torque. This is a distinct advantage of series wound motors, especially in traction motor applications. It should be noted that this equation is only valid up until the point where the magnetic circuit becomes saturated, after this point the field strength becomes constant and the relationship between current and torque becomes linear. This relationship is illustrated in figure 2.4.

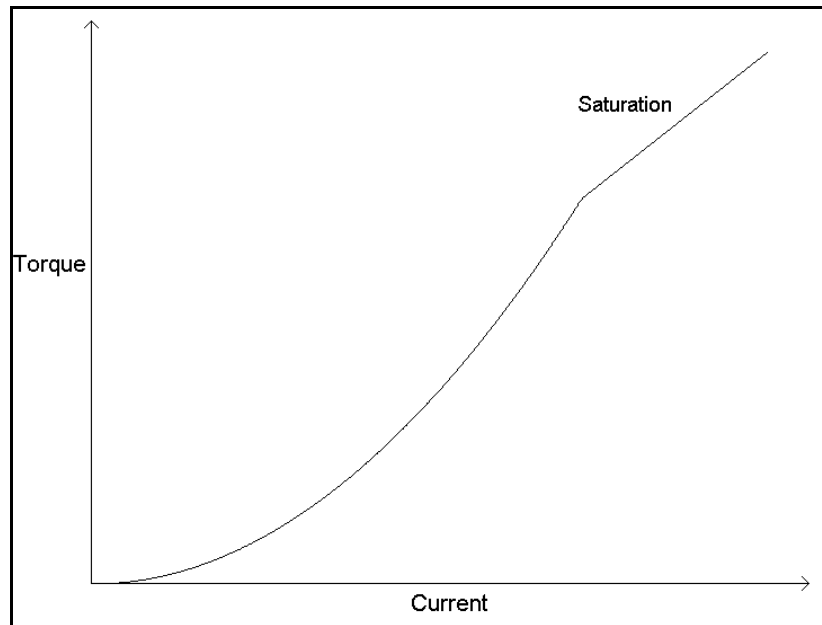


Figure 2.4 – Series Wound Motor Current-Torque Curve

#### 2.1.4 Speed

In all DC motors, speed is proportional to supply voltage except for the effect of the armature resistance. For series wound motors, this relationship is given by:

$$\omega_m = \frac{V}{\sqrt{kT}} - \frac{R_a}{k}$$

A graphical representation of the relationship between torque and speed is given in figure 2.5. By examining the graph, it can be seen that a series wound DC motor with no load torque will attempt to increase speed to infinity! Of course, this is impossible in practice as there is always some frictional load on the motor due to wind and bearing resistance. However, it is still possible for an unloaded series DC motor to reach dangerous speeds with the potential to damage the motor. Thus there must be some form of over speed protection built into systems using series wound motors.

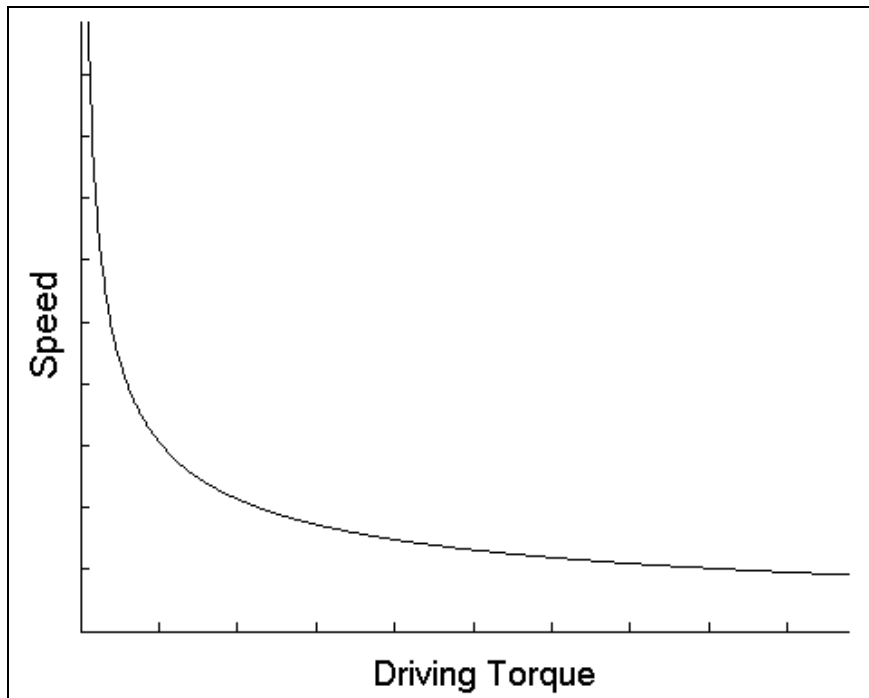


Figure 2.5 – Series Wound DC Motor Torque-Speed Curve

### 2.1.5 Power and Efficiency

The power a motor is drawing from the supply is given by:

$$P_{in} = VI$$

The mechanical power produced by the motor is given by:

$$P_{in} = T\omega$$

There are two extreme operating points, when there is no load on the motor and when the load is large enough to stall the motor. When there is no load on the motor (i.e. mechanical power = 0), the only power drawn from the supply will be what is required to overcome the losses within the motor. When the motor is driving a very large load, such that the rotor is stalled (speed = 0), the mechanical power is also zero but the input power will be very large. This is because at zero speed, the back EMF will be zero, therefore the power drawn from the supply will be  $P = V^2 \div R$ .

Efficiency is defined as the amount of input power divided by the actual output power. That is;

$$efficiency = \frac{P_{out}}{P_{in}} = \frac{T\omega}{VI} \times 100$$

## Section 2.2 – DC-DC Converter Operation

A DC-DC converter allows DC voltages to be raised or lowered similar to how a transformer does with AC voltages. DC-DC conversion became very popular with the advent of power transistors, and since then many circuits have been invented. There are countless types of DC-DC converter circuits but they all operate on the same basic principles. The simplest types of converter are the non-isolated buck and boost converters. Other types include the Ćuk converter and the Buck-Boost converter, as well as isolated designs such as the forward converter and flyback converter.

### 2.2.1 Switch-Mode Buck and Boost Converter Circuits

The basic circuit configuration used in the buck converter is shown in figure 2.6.

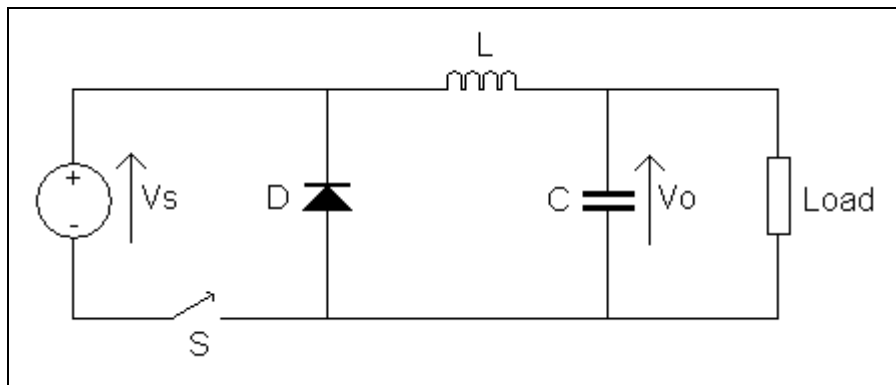


Figure 2.6 – Buck Converter Circuit

There are only four main components to the buck converter circuit: a high speed switching device (S), a flywheel diode (D), an inductor (L) and an output filter capacitor (C). The output voltage is monitored and maintained at a desired level by a control circuit which switches S on and off at a set frequency, but with a varying duty cycle. The duty cycle is defined as the ratio between on-time to the period of the switching frequency. When the switch is turned on current begins flowing from the supply through L, then into C and the load. The inductor builds up energy in its magnetic field, with the voltage drop developed across L opposing (bucking) some of the input voltage. When the switch is turned off, by its nature the inductor opposes any drop in current by suddenly reversing its voltage and supplies current to the load via the diode. The DC output voltage which appears across the load is a fraction of the input voltage, and is proportional to the duty cycle. The output voltage can be defined as:

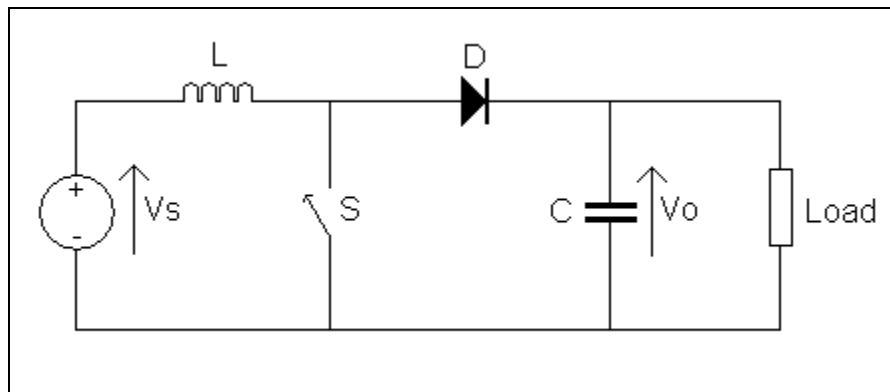
$$V_o = V_s D \quad \text{where } D \text{ is the duty cycle}$$

From this equation, it can be seen that varying the duty cycle will allow the output voltage to be controlled. In addition, just like a transformer the ratio of output current to input current is the same as the reciprocal of the voltage ratio - ignoring losses. This means that:

$$P_{out} = P_{in}, \text{ so } \frac{V_s}{V_o} = \frac{I_o}{I_s}$$

This means that if the circuit is stepping down the voltage by half, the input current is only half the value of the output current.

Figure 2.7 shows the basic boost converter circuit. It uses the same components as the buck converter but they are arranged differently.



**Figure 2.7 – Boost Converter Circuit**

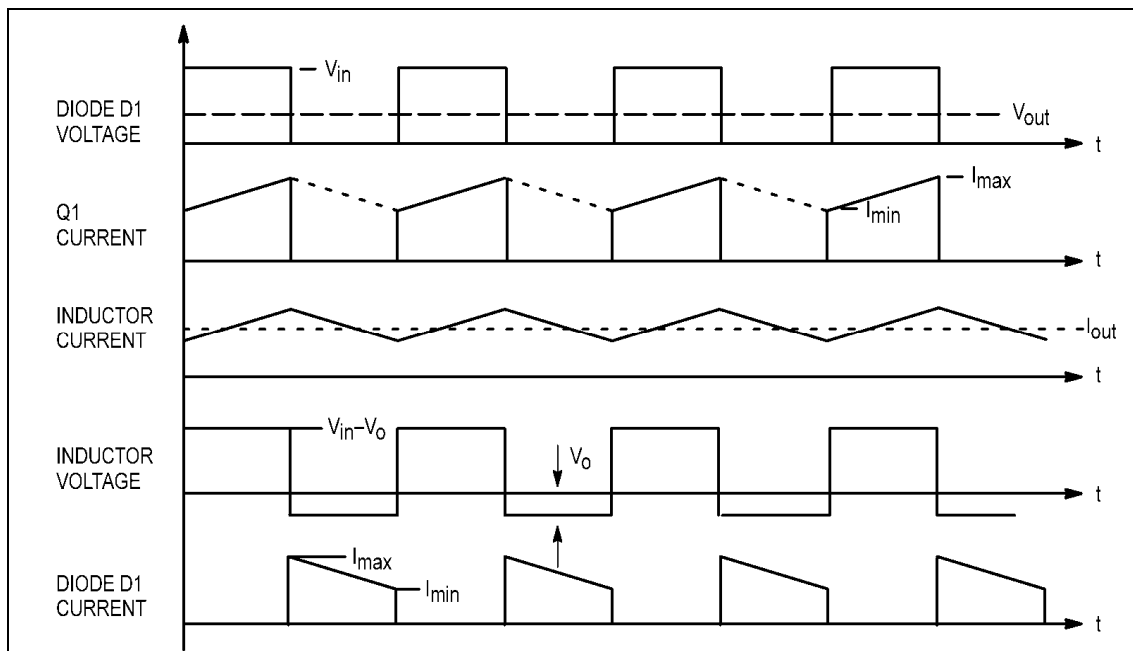
Once again the duty cycle of the high speed switch controls the output voltage. The circuit boosts the voltage using energy stored in the inductor. When the switch is turned on, current flows from the supply through L, and energy is stored in the inductor's magnetic field. At this stage there is no current through the diode, the load current is supplied only by the charge in C. When the switch is turned off, L opposes the change in current by reversing its voltage. In trying to maintain the current flow, the inductor voltage raises so that the inductor voltage adds to the source voltage. Current due to this boosted voltage flows from the source through L, D and the load, recharging C as well. This time the output voltage is related to the duty cycle by:

$$V_o = \frac{V_s}{(1-D)} \quad \text{where } (1-D) \text{ is the proportion of the switching cycle that S is off.}$$

Assuming no losses,  $P_{out} = P_{in}$ . As would be expected, a doubling of the output voltage would cause a doubling of the input current.



The waveforms depicting the voltage across the inductor and the current through it are shown in figure 2.8. These waveforms are similar to those that should be observed across the motor during both buck and boost modes.



**Figure 2.8 – Continuous Mode Voltage and Current Waveform of a Buck Converter**

### 2.2.2 Using a DC Motor as a Load

Using a DC motor as the load in a DC-DC converter can simplify the circuit somewhat. Since the motor is highly inductive due to the many turns of copper wire in its armature and field coils, it can be used as the inductor in the DC-DC converter circuit. In addition to this, an output capacitor is also not required to smooth the output voltage. This is because even if the input voltage is not ripple-free the motor speed will still remain constant due to the inertia of its rotating mass.

There are some detrimental effects of using DC-DC converters to drive a motor. These stem largely from the high  $dv/dt$  rate created by the motors inductance when it is switched on and off. This can cause stress on the insulation of the windings and excess heating of the motor. The higher the switching speed, the more often there are  $dv/dt$  events and the more stress on the motor. How much of an effect this has in practice largely depends on the construction of the motor and the quality of its insulation. Using a lower frequency also helps reduce losses in the power electronics. Normally the switching frequency is chosen so that it is out of the human hearing range, but not any higher than necessary. A frequency of around 15 kHz is typical for a DC motor control.

### 2.2.3 Pulse Width Modulation

Pulse width modulation is used to control the switches in switch-mode DC-DC converters. In a motor control application, PWM provides a way of controlling the average current or average voltage delivered to the motor simply by switching the transistors on and off in rapid succession

For a simple example, imagine a 100W light bulb connected to a switch. If you turn on the switch for 10 seconds, then turn it off again the average power over those ten seconds was 100W. However, if you turned it on for 5 seconds, then off another five then the average power over 10 seconds was only 50W. This is how PWM controls the motor, just at a much higher switching speed. The motor is switched on and off thousands of times a second with the duty cycle controlling how much current is delivered to the motor. The switching is so fast that it appears to the motor (due to its inductance) that it is receiving a steady voltage. Therefore, the motor's output torque can be varied smoothly by controlling the duty cycle of the square wave PWM signal. In a similar example to the light bulb situation given earlier, take a motor that will spin at 1000 r/min at 12V and 10A. To achieve half of this speed, a PWM signal with a duty cycle of 50% will be applied to the motor. This will deliver 12V at 10A to the motor for 50% of the time. This way, the average speed of the motor will be 500 r/min.

## Section 2.3 – The Selected Power Electronic Configuration

The typical way to get four quadrant operation using electronics is to use four switching devices connected in the H bridge configuration illustrated in figure 2.9. Switching on S1 and S4 will apply voltage across the motor in one polarity, and S3 and S2 the other. The trouble with this circuit is that it doesn't work for series wound motors as is. This is due to the fact that the field is in series with the armature. If the voltage across the motor is reversed, it will reverse the voltage across both the armature and the field, so the motor will continue spinning the same way! This is the reason why series motors can also work on AC. To circumvent this problem, either the field or the armature connections must be reversed, but not both. Another problem with this configuration is that it requires four high power transistors (or four banks of smaller transistors), which are expensive. One of the aims of this project is to produce an economical design, so a more cost-effective approach must be taken.

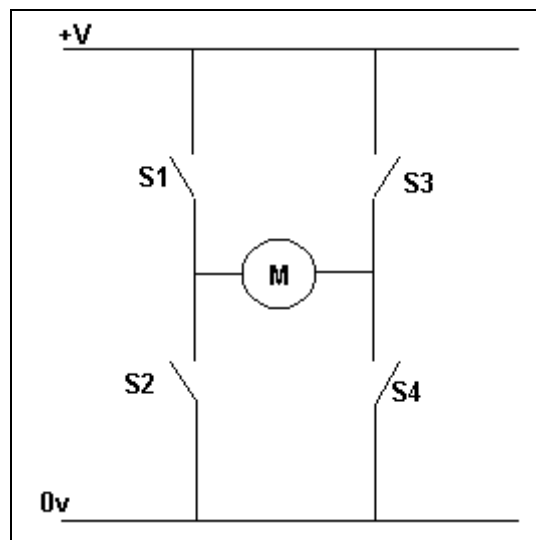


Figure 2.9 – H Bridge Configuration

### 2.3.1 Economical Four Quadrant Circuits for Series Wound Motors

The solution can be found in the half bridge, and some large mechanical relay switches (contactors) to reverse the field connections. This circuit, shown in figure 2.10, allows full four quadrant operation while using half of the expensive power transistors required by the full bridge. The polarity of voltage and current for each of the four quadrants is graphically illustrated in figure 2.11. It may at first seem that the use of contactors will make the design more complicated and prone to contactor faults. However, the field connections need only be switched when the motor must be reversed. This means that the motor will be at a stop when they operate, and thus they are not interrupting any current. This will significantly prolong the life of the contactors.

In addition, the time taken for the contactors to switch is also irrelevant. The time it takes for the contactors to switch is insignificant when compared to the time that must be allowed for the motor to slow to a stop before its direction can be reversed.

This circuit also allows the field current and armature current to be equal. This is important, because these currents must be the same to minimise arcing of the commutator. Otmar Ebenhoech (a motor control designer), in consultation with Eric Dieroff (an engineer at Advanced DC Motors) found that “there is an area on the commutator where adjoining commutator bars have very little voltage potential between them. The motor's brushes are located here, to minimize arcing and heating, and meltdown or brush failure. Failure to keep equal currents in the two windings causes this critical dead zone to move. The brushes now short out bars with a higher differential voltage. Sparking results. Excessive arcing can extend all the way to the opposite polarity brush, creating a fireball effect.” (Ebenhoech 1998)

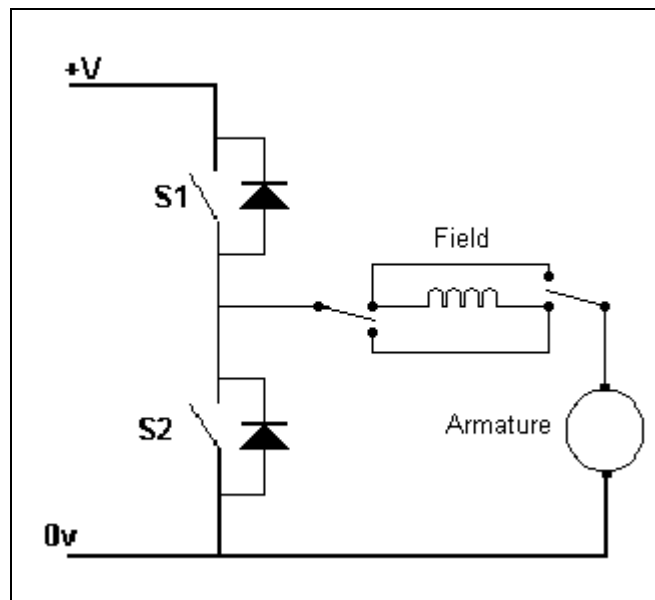


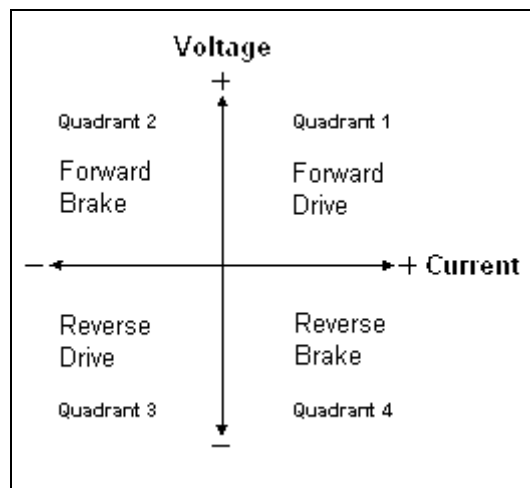
Figure 2.10 – Half Bridge with Reversing Contactors

### 2.3.2 How the Half-Bridge Circuit Works

This circuit effectively consists of two types of DC-DC converter combined. There is a buck (step-down) converter and a boost (step-up) converter. The buck converter is used to step the battery voltage down to control the speed of the motor. The boost converter is used to boost the motor voltage high enough to charge the batteries when the circuit is operating in the regenerative mode. The PWM technique is used to control the switches in the half bridge circuit to vary the power delivered to the motor. So how does this circuit achieve four quadrant operation? The top switch of the bridge (S1) and the bottom diode form a step down converter that can reduce the voltage to the motor.

When S1 turns on, current flows from the battery into the field and armature of the motor, causing it to spin. When it turns off, the current in the motor must continue to flow as the motor is very inductive, so it does so through the lower diode. Thus, varying the duty cycle of S1 will control the motor in the motoring mode. To reverse the rotation of the motor, the connection of the field is simply reversed using the contactors.

Operating the motor as a generator is slightly more complex. To use the motor as a generator the current must flow out of the motor into the batteries. For this to happen, the voltage across the motor must be greater than the battery voltage. This physically cannot happen unless the motor voltage is boosted somehow. This is the role of S2; it forms part of a step-up converter. To make the circuit work as a boost converter, S2 is turned on, which effectively shorts out the motor. This causes a large current to build up in the motor loop. When this current has built up, S2 is turned off. Due to the inductive nature of the motor, it will try to keep current flowing, so its voltage will rise until it becomes high enough for current to flow through the upper diode and into the battery pack. Applying the PWM control signal to S2 allows the motor to be controlled as a generator. Once again, to perform regenerative braking while the motor is reversing, the contactors are used to reverse the field connection.



**Figure 2.11 – Four Quadrants of Operation**

## Section 2.4 – Gate Drive

As stated previously, it was decided that power N-Channel MOSFET type transistors would be used in the power stage of this design. These devices have three terminals: a gate, drain, and source. The MOSFET is switched on by applying a voltage to the gate terminal. With the N-type MOSFETs used in this design, this normally must be about 10V greater than the source potential. This allows current to flow from the source to the drain. In this way, the MOSFET acts as switch. In fact it is one of the closest semiconductor equivalents to a real mechanical switch. Therefore, for the electronic control circuit to switch the MOSFET on and off there must be some form of “gate driver” circuit.

### 2.4.1 What is a Power-MOSFET Gate Driver?

A gate driver is a power amplifier that accepts a low-power input from a controller IC and produces the appropriate high-current gate drive for a power MOSFET (Davis 2004, p. 1).

A gate driver is used when the pulse width modulation (PWM) controller cannot provide the output current and voltage required to drive the gate capacitance of the associated MOSFET. Gate drivers can be constructed from discrete components (transistors and sometimes transformers). Many semiconductor manufacturers also provide dedicated drive ICs that contain most of the components required to drive the MOSFETs and often extra features such as protection logic and voltage level converters to drive high-side MOSFETs.

The MOSFETs with the highest current ratings are of the N-channel type, these are also the most economical. However, to fully turn these MOSFETs on their gate voltage must be 10V greater than the voltage at the source terminal. This voltage is easily provided for the low side switch (S2 in the circuit shown previously) as it is just 10V. For the high side (S1) the voltage must be 10V greater than the high side MOSFETs source terminal, this is a little more difficult to provide. It can be done with an external power supply designed to produce a sufficiently high voltage, or with a circuit designed to boost a lower voltage. The usual way to do this is with a bootstrap capacitor and a level shifting circuit or using a charge-pump circuit. Most of the dedicated half or full bridge driver ICs have these circuits built in. This helps simplify gate drive circuits and make them more robust.

## 2.4.2 Circuit Model for a Power MOSFET Gate

The basic circuit model of a Power MOSFET gate is given in figure 2.12. The model includes the major parasitic components that influence high-speed switching times of the MOSFET. Namely, these are:

- The gate-to-source capacitance ( $C_{GS}$ )
- The gate-to-drain capacitance ( $C_{GD}$ )
- The drain-to-source capacitance ( $C_{DS}$ ), and
- The internal gate resistance ( $R_G$ )

The value of the source inductance ( $L_S$ ) and drain inductance ( $L_D$ ) depends on the design of the MOSFET's package.

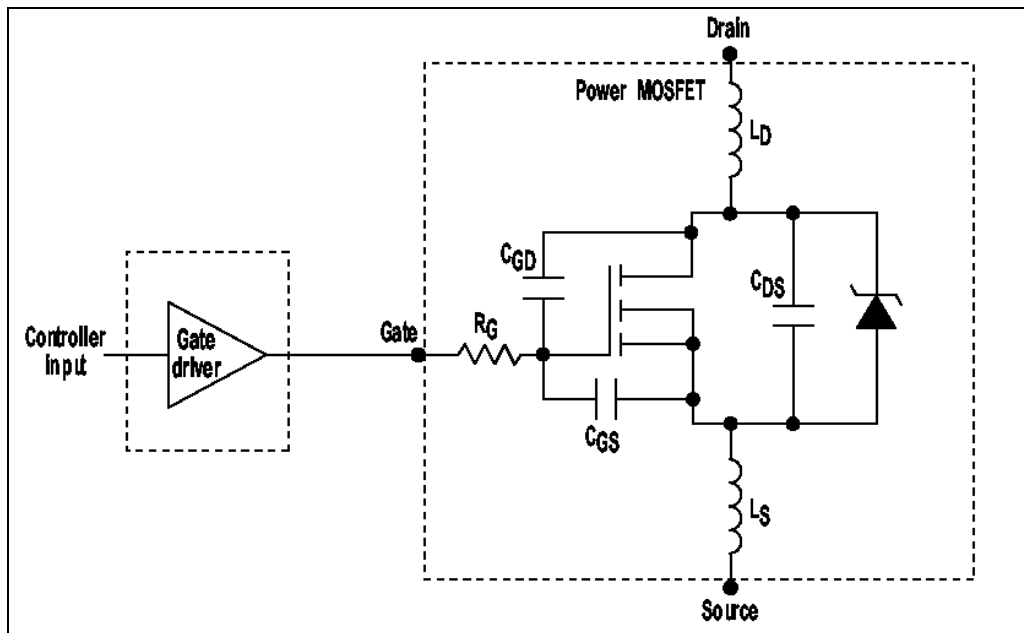


Figure 2.12 – Simplified MOSFET Model and Gate Driver (Davis 2003)

MOSFET devices have what is known as “Miller capacitance” between their gate and drain terminals. In low voltage or slow switching speed applications this gate-drain capacitance does not normally present a problem, however when high voltages are switched quickly it has a significant effect and must be taken into consideration.

## 2.4.3 Gate Driver Design Considerations

In this application, it is only desirable to operate the MOSFETs in one of two states, either fully on or fully off. In between these two states, the MOSFET is operating in the linear region. In this region its on resistance is not low enough for good conduction and the MOSFET will generate a lot of power as  $I^2R$  loss. This is very undesirable, as not only is it inefficient, but also causes excess heating of the transistor.

Unfortunately, it is not possible to totally avoid this linear region due to the parasitic components explained above. The device can not instantly switch from off to on or vice-versa, the capacitances and resistances present slow down the switching time as they form an RC circuit with a time constant. However, with good gate driver design it is possible to minimise the switching time, hence keeping losses within a manageable level.

An important attribute for the gate driver is its ability to provide sufficient drive current to quickly pass through the Miller Plateau Region of the MOSFET's switching transition (Davis 2003). This interval occurs when the transistor is being driven on or off, and the voltage across its gate-to-drain parasitic capacitor ( $C_{GD}$ ) is being charged or discharged by the gate driver.  $C_{GD}$  is much smaller than  $C_{GS}$ , but its value is multiplied by the gain of the MOSFET (called the Miller effect). When switching high voltages, the gate-drain Miller-effect capacitance greatly exceeds the gate-source capacitance.

Figure 2.13 graphs the total gate charge as a function of the gate drive voltage of a power MOSFET. Total gate charge ( $Q_G$ ) is how much must be supplied to the MOSFET gate to achieve full turn-on. It is usually specified in nanocoulombs.

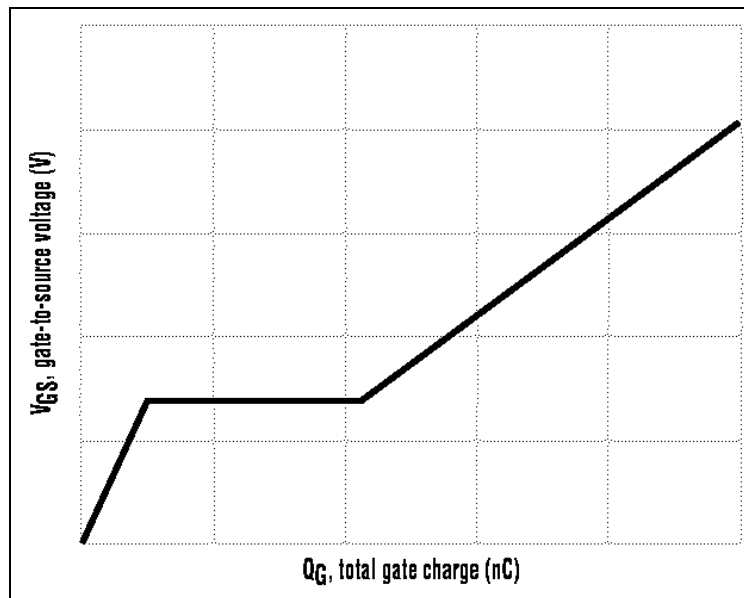


Figure 2.13 – Power MOSFET Gate Drive Characteristics (Davis 2003)

Another issue caused by the Miller Capacitance exists in the half bridge circuit. The problem occurs when the top MOSFET turns on, causing the drain voltage of the bottom device to rise very quickly. This high  $\Delta V$  rate capacitively couples to the gate of the bottom MOSFET via the Miller Capacitance. This can cause the gate voltage of the bottom MOSFET to rise resulting in turn on of this device as well! This creates a “shoot-through” condition and a high probability of instant MOSFET failure.



This consequence of the Miller Effect can be avoided by using a very low impedance gate drive which clamps the gate voltage to 0 volts when in the off state. This reduces the effect of any spikes coupled from the drain. Further protection can be gained by applying a negative voltage to the gate during the off state.

#### **2.4.4 Dedicated MOSFET Driver ICs**

There is a plethora of MOSFET gate driver ICs made by manufactures such as Maxim, Intersil and National Semiconductor. The most common bridge drivers are the HIP2101 (half bridge) and HIP4081 (full bridge) from Intersil. The original Intersil drivers cannot supply a very high gate drive current (around 2A), and many other manufacturers have designed equivalent pin-compatible chips with higher output currents.

One such gate driver is the LM5101A from National Semiconductor, and was chosen for this design. The LM5101A is a 3A (5A peak) high-voltage gate driver for the high-side and low-side N-channel MOSFETs in a synchronous buck or bridge configuration. The floating high-side driver can operate with supplies up to 100V. Outputs are independently controlled with TTL-input thresholds. Both the low-side and the high-side power rails include undervoltage lockout. Turn-off propagation delay is typically 25ns, and the ICs can drive 1000pF loads with 15ns rise and fall times.

## Section 2.5 – Software Requirements

Since the design requires the use of a microcontroller, software (also known as firmware) must be written that performs the necessary functions of an adjustable speed drive. The software that is programmed into the microcontroller has control over all aspects of the adjustable speed drive and it is important that it is stable and bug free to prevent unreliable or even dangerous operation.

### 2.5.1 What is the purpose of the software?

The basic task of the software is to control the microcontroller's on-chip PWM module based on accelerator/decelerator position and motor current. Most of the inputs to the microcontroller are via its on-chip 10bit ADC. The main tasks of the software can be grouped into three functions;

- Programming mode – when the microcontroller is linked to a PC to set the drive parameters.
- Motoring mode – reads forward/reverse selector and accelerator pedal position and controls motor current in accordance by using the buck converter, while ensuring the preset current and speed limits are not exceeded.
- Regeneration mode – reads forward/reverse selector and decelerator pedal position and controls the boost converter to deliver power to the batteries. Once again, the preset limits must not be exceeded and the batteries must not be overcharged.

The software must operate in such a way as to ensure predictable and safe operation of the drive. This means that failsafes are required to ensure that the event of a software error or crash does not have a dangerous effect on drive operation.

## **Section 2.6 – Protection**

Protection is a very important aspect of any adjustable speed drive. It is necessary to ensure that at no time the drive operates in a way that may damage the electronics, the motor, its associated load or even the operators of the device itself. There are many forms of protection, those applicable to this design include:

- Motor current limiting
- Motor speed limiting
- Battery over and under voltage limiting
- Thermal limiting for the power electronics and/or motor
- Fault detection to help prevent undesired operation or damage to equipment

It can be seen that protection can take two general forms, either limiting some parameter within a specified range or detecting faults and taking some corrective action. To maintain the simplicity and low cost of the circuit, much of the protection will be accomplished using software. While this may not always be best practice, in this application there should always be fuses and kill switches as part of a vehicle's system. In the event of a catastrophic failure, these devices can be used to shut down the system.

### **2.6.1 Limiting**

There are various parameters that require limiting to ensure that the operation of the adjustable speed drive remains within safe bounds. The various type of limiting incorporated in this design are listed in the points above. These are; motor current limiting in all four quadrants of operation, motor speed limiting in forward and reverse, battery over/under voltage sensing and thermal protection for the electronics. This limiting will be performed by the software, with the analog sensors read by the microcontroller's built in ADC.

## 2.6.2 Failsafes and Fault Detection

In addition to limiting, it is also important to detect faults that may cause the drive to function erratically. The following features will be implemented in the software:

- Do not reverse motor unless it's speed is zero
- Do not switch on main contactor until ripple capacitors are fully pre-charged
- Decelerator pedal overrides accelerator pedal
- Do not switch on power stage if the battery voltage is too high or too low
- Do not attempt to perform regeneration if the battery voltage is too high (i.e. the batteries are fully charged)
- Do not allow both high side and low side transistor to turn on at once as this will instantly destroy them (also include hardware logic to prevent this).

The reasons for most of these features are fairly self explanatory, for example, the motor must not be reversed while it is still spinning, or if the batteries are flat or connected incorrectly (over/under voltage) the electronics must shut down to prevent damage. The reason for pre-charging the very high capacitance filter capacitors via a resistor before switching on the main contactor is to prevent damage to the main contactor from the high inrush current that would otherwise occur if the discharged capacitors were instantly connected to the battery pack. It also helps to detect if there is a short in the high-side of the power stage, which would cause the vehicle to instantly try to move at turn on. In this situation the capacitors would never charge because the motor would be in circuit via the shorted MOSFET, dropping all the battery voltage across the pre-charge resistor. The microcontroller could detect this fault and display an error. The various protection circuits and how they are implemented are discussed further in future chapters.

# Chapter 3

## **3.0 Prototyping and Testing of the Design Concepts**

This chapter deals with the prototyping and testing of the basic circuit blocks that form the entire circuit prior to full system implementation. It discusses the parts chosen, why and how they are used, and their relevance to the whole system.

### **Section 3.1 – Microcontroller Modules**

The microcontroller chosen for this design comes from the PIC16F range by Microchip. The particular chip chosen is the PIC16F877A – a 40 pin, 8 bit 20MHz RISC device. It was chosen because of its low cost, high feature count, software support, and availability. The microcontroller has most of the required functions built in. It includes 8k of program memory, an 8 channel 10-bit ADC, 2 channel 10bit PWM module, and a hardware UART (for serial communication), all of which are used in the design. An extract from the data sheet listing the devices' features and pin out is given in Appendix B.

The development environment chosen to write the software for the microcontroller was Mikrobasic, by Mikroelektronika. It is a simple to use program that supports many devices and Mikroelektronika offer a demonstration version of their software which can be obtained as a free download from the internet. Mikrobasic also includes many “libraries” of code which provide easy access to the microcontrollers' features. These include libraries for the PWM, ADC, EEPROM and USART, as well as libraries for external devices such as LCD displays and the I<sup>2</sup>C bus. Other valuable libraries provide mathematical features such as floating point operations and trigonometric functions.

#### **3.1.1 Programming the microcontroller**

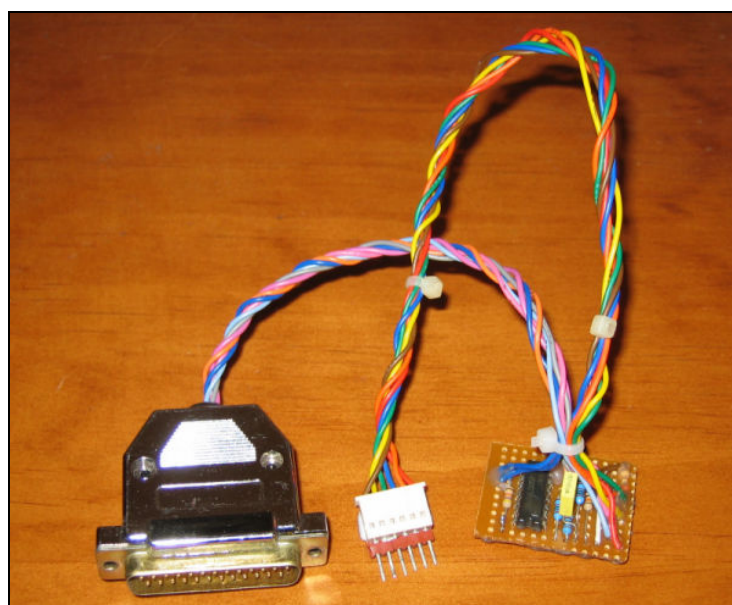
“Programming” the microcontroller refers to loading the firmware into the microcontroller from the development system. The “firmware” is the software that is written for the microcontroller to run. The PIC16F series of microcontrollers can be programmed in two ways. One is known as “high-voltage” programming, as it uses 12V to program the internal flash ROM.

To program the PIC in this way, the chip must be removed from the circuit and placed into a special programmer board that interfaces to the development PC via a serial or parallel port. This has the disadvantage of requiring the removal of the microcontroller from the circuit for programming but it allows access to code protection features of the chip.

The alternative method is known as “low-voltage” programming or in-circuit (ICSP) programming. This method allows the PIC to remain in circuit during programming, which can greatly speed up the prototyping stages. However, it does require the use of an additional pin on the microcontroller and prevents some of the chips code protection features from being accessed. This is the method of programming chosen for this design, due to its simplicity and ICSP capability.

Programmers for the PIC range of microcontrollers can be purchased from many outlets, however due to their relatively simple construction it was decided that one would be built from scratch to save costs.

A freeware programmer by Christian Stadler was found on the internet. It included a schematic for a simple low-voltage ICSP circuit and Windows programming software. The schematic for the programmer is given in Appendix C. The circuit was constructed on a piece of Veroboard, with the appropriate connectors attached on flying leads. A DB25 plug connects to the printer port of the PC, and the programmer connects to the circuit via a 6 pin SIL header. It is powered by the circuit it is connected to. A picture of the programming hardware is shown in figure 3.1.

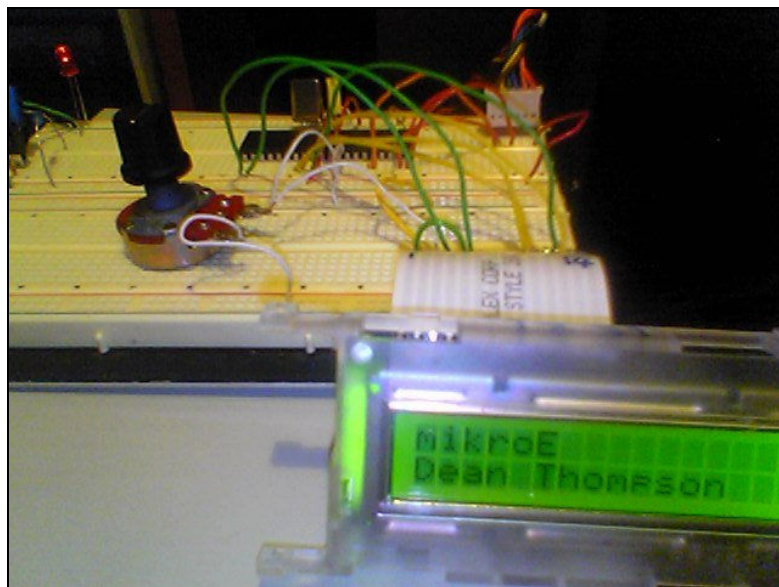


**Figure 3.1 – PIC Programmer**

The programming software is able to take the HEX file output of the Mikrobasic compiler and transfer it into the PIC's flash memory via the PC's printer port. The programmer can be left connected to the circuit, which makes programming quick and simple thus allowing for faster prototyping. The flash memory can be reprogrammed thousands of times, so the prototype's software can be changed and tested many times without needing to buy new microprocessors.

### 3.1.2 LCD

A character LCD display is incorporated into the design to give the operator information while the drive is in operation as well as convey diagnostic information during the testing stages. The chosen LCD screen can display two rows 16 characters long. It uses the standard Hitachi HD44780 LCD chipset, which is supported by a built-in library in the Mikrobasic development environment. This makes writing the firmware much simpler. The LCD and microcontroller were connected together on a prototyping board via a 4 bit interface to test their communication. The LCD worked as expected and no difficulties were encountered interfacing the LCD. A test message displayed on the LCD is pictured in figure 3.2.



**Figure 3.2 – Testing the LCD**

The LCD is very useful during the design and prototyping stages, as it is able to give a real-time display information which can be used in debugging the firmware. In the completed design, the display will be used to show parameters such as motor speed, motor current, battery voltage and other useful status information.

### 3.1.5 ADC

The analog-to-digital converter is a very important part of the design. It is integral to the operation of the entire circuit. The ADC will be used to measure the accelerator and decelerator pedal positions, motor current, battery voltage, motor speed and power stage pre-charge voltage. There are many different ways of performing an analog to digital conversion. The main types of converter are listed below.

- Successive Approximation Converter - A successive approximation converter provides a fast conversion of a momentary value of the input signal. It works by first comparing the input with a voltage which is half the input range. If the input is over this level it compares it with three-quarters of the range, and so on. Twelve such steps give 12-bit resolution. While these comparisons are taking place the signal is frozen in a sample and hold circuit. After A-D conversion the resulting bytes are placed into either a pipeline or buffer store. A pipeline store enables the A-D converter to do another conversion while the previous data is transferred to the computer.
- Dual Slope Integrating Converter - This converter reduces noise but is slower than the successive approximation type. It lets the input signal charge a capacitor for a fixed period and then measures the time for the capacitor to fully discharge at a fixed rate. This time is a measure of the integrated input voltage, which reduces the effects of noise.
- Charge Balancing Converter - The input signal again charges a capacitor for a fixed time, but in this converter the capacitor is simultaneously discharged in units of charge packets: if the capacitor is charged to more than the packet size it will release a packet, if not a packet cannot be released. This creates a pulse train. The input voltage is determined by counting the pulses coming out of the capacitor. Noise is reduced by integrating the input signal over the capacitor charging time.
- Flash Converter - A flash converter is one of the fastest types of converter. Like the successive approximation converter it works by comparing the input signal to a reference voltage, but a flash converter has as many comparators as there are steps in the comparison. An 8-bit converter, therefore, has 256 comparators.



- Sigma-Delta Converter - This converter digitises the signal with very low resolution (1-bit) and a very high sampling rate (MHz). By oversampling, and using digital filters, the resolution can be increased to as many as 20 or more bits. Sigma-delta converters are especially useful for high resolution conversion of low-frequency signals as well as low-distortion conversion of signals containing audio frequencies. They have good linearity and high accuracy.

The microcontroller's built in 8 channel ADC has 10 bit accuracy. It is a successive approximation type converter. Two of these eight channels can be also used to provide an external voltage reference for the ADC. This gives the opportunity to use a more accurate voltage reference, but sacrifices two of the channels. In the case of this design, an external reference will be used to improve the accuracy of the motor current readings. The ADC has an acquisition time of around 19 $\mu$ s, the actual time depends on the input impedance and temperature. This is the time it takes to charge the ADC's holding capacitor to the input voltage level. This acquisition time must be provided if the ADC is to meet its specified accuracy. Microchip provides a formula to calculate the acquisition time, which is;

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$

$$= 2\mu\text{s} + C_{HOLD} (R_{IC} + R_{SS} + R_S) \ln(1 \div 2047) + ((Temp - 25^\circ C) 0.05\mu\text{s} / ^\circ C)$$

where  $R_{IC}$  is the internal interconnect resistance ( $\leq 1k$ ),  $R_{SS}$  is the sampling resistance (7k)  $R_S$  is the source impedance and  $C_{HOLD}$  is the hold capacitance (120pF) (Microchip 2003).

The time it takes to perform an A to D conversion depends on the clock frequency of the microcontroller. In this case, it is operated at its maximum frequency of 20MHz. This gives a conversion time of 1.6 $\mu$ s. This is the minimum conversion time possible for the PIC16F877A. Since the conversion time is much faster than the acquisition time, care must be taken when writing the software to ensure that the holding capacitor is charged before taking a measurement.

The libraries built into Mikrobasic take care of setting the correct registers in the microcontroller to operate the ADC, this helps to speed software development time. The ADC was tested on a prototyping board by connecting a potentiometer to one of its inputs and displaying the measured voltage on the LCD. The reading was checked with a digital multimeter and found to be accurate to within around 0.05 volts. A sample of the LCD's output is illustrated in figure 3.3.



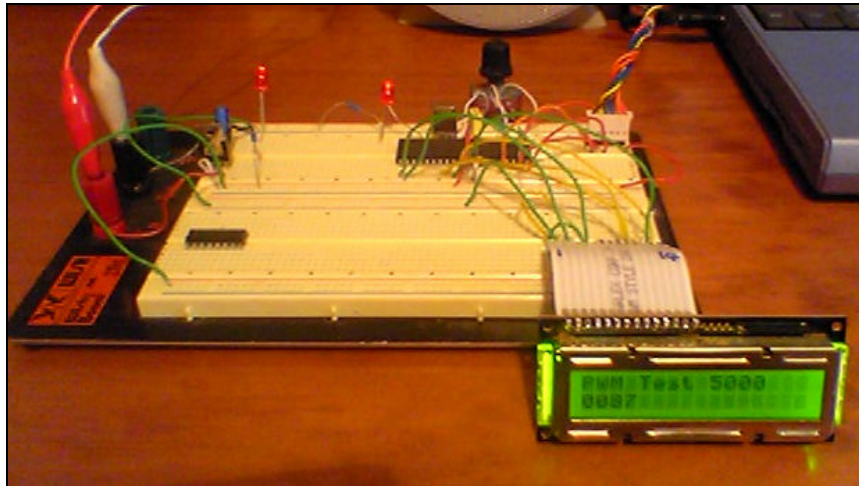
**Figure 3.3 – ADC Testing**

#### **3.1.4 PWM**

The PWM module is another device that is integral to the design; it provides the PWM signal to drive the transistors in the power electronics. The PWM module in the PIC17F877A is able to produce two PWM output signals. For this design, only one of the PWM channels is used. A Pulse Width Modulation output (Figure 3.4) is a signal that has a period and a time that the output stays high (duty cycle). The period is the duration after which the PWM rising edge repeats itself. The resolution of the PWM output is the granularity with which the duty cycle can be varied. The frequency of a PWM is simply the inverse of the period ( $1/\text{period}$ ). Each CCP module of the PIC can support one PWM output with minimal software overhead. This PWM signal has a resolution of up to 10-bits, this gives 1024 steps of variance, which corresponds to a maximum accuracy of  $T_{\text{osc}}$  (50ns when the device is operated at 20 MHz).

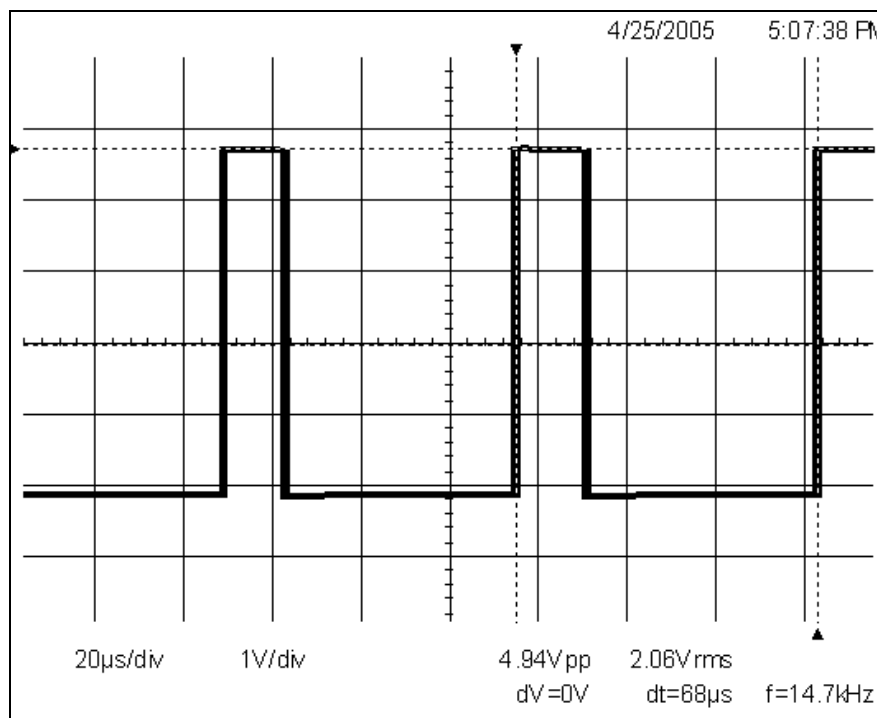


A circuit to test the PWM module was built on a prototype board. It reads the position of a potentiometer via the ADC and adjusts the PWM duty cycle from 0-100% accordingly. The PWM signal is used to vary the brightness of a LED and the PWM frequency and duty cycle is displayed on the LCD screen. A photograph of the test circuit in operation is presented in figure 3.6.



**Figure 3.6 – PWM Test Circuit**

The PWM output was monitored by an oscilloscope to ensure the duty cycle varied according to the position of the potentiometer. Figure 3.7 shows the 15kHz PWM signal when the potentiometer is at about 20% of its travel (i.e. 20% duty cycle).



**Figure 3.7 – Microcontroller's PWM Output Signal**



## Section 3.2 – Analog Circuits

This section discusses the analog circuits that interface to the ADC of the microcontroller. Design challenges are to be expected in the area of the circuit due to the level of noise present from the power electronics and the digital microcontroller. The analog inputs to the microcontroller consist of motor current, motor speed, battery voltage, power stage voltage, accelerator and decelerator pedals and the analog voltage reference.

### 3.2.1 Motor current

The motor current is the most critical input to the microcontroller, the reading must be accurate and fast enough to be able to protect the power electronics from excess current which could easily cause damage to the output MOSFETs. Unfortunately, motor current is also difficult to measure, due to the large currents and fast transients present.

The classic approach to current sensing in a motor's electronic power stage is to use a current sensing resistor (called a shunt). Because of the need for electrical isolation, other current sensors are sometimes employed. These include Hall current sensors, current transducers, and current transformers (Valentine 1998). The first three of these are suitable for measuring DC current.

Hall-effect current sensors are able to measure the current flowing through a conductor over a wide frequency range, from DC to over 10kHz. They can be used to determine the current flowing by measuring the magnetic field created by the flowing current. Thus, they are able to provide voltage isolation and are most suitable for high voltage circuits (>50V). Additional circuitry is required to amplify the output from the Hall sensor to a useful level. The main limitation of Hall sensors is their lack of sensitivity and stability over temperature.

A current transducer is a form of current sensor with built-in electronic interface that provides a digital or analog output signal. These are very simple to use as there is no need to design interface and op-amp circuits as these are included as part of the device. However, these devices are very expensive and not widely available.

Resistor current sensing is usually used for lower power motor controls and those that are operated at low voltages (<50V). It is a simple and cheap way to measure current. It is achieved by placing a resistor in series with the current to be measured, and reading the voltage developed across the resistor.

Special resistors are designed for this purpose, these resistors are called shunts and have low inductance to minimise ringing in PWM-type circuits and low resistance to minimise  $I^2R$  loss. The resistance of a shunt is typically in the hundreds of microohms, and usually rated to produce a voltage in the hundreds of millivolts. Op-amps are used to amplify the voltage developed across the resistor to a useful level. Because of the low voltage output of the shunt and the consequently high gain amplifier required, problems can arise with EMI and noise. To reduce these problems, the amplifier circuit should be placed close to the shunt and kept as much away from high current circuits as possible. High frequency filters can also be added to the amplifier's input. The timing of the current measurement can be arranged so that it takes place just after the power stage is switched on and before it is switched off, this will minimise interference due to electrical noise from the power stage's switching transitions.

The major drawback to resistor current sensing is the lack of voltage isolation. This can be overcome by using an optocoupler, a linear optocoupler for current sensing or an optocoupler on the output of a comparator (to detect overcurrent conditions only). Due to the low voltages seen in this project, the requirement for isolation is not critical and it is expected that a non-isolated resistor current sensing circuit will suffice. A 50mV shunt is placed in the high-side of the power stage to measure the motor loop current.

Since this motor control is capable of regeneration, current must be able to be measured in both directions. To simplify the current sensing circuit and speed up design time, it was decided that instead of designing an amplifier circuit from discrete components a special function IC from Maxim would be used. The particular chip chosen is a MAX4081SASA, which is a 76V high-side bi-direction current sense amplifier with a gain of 60V/V, the circuit used is shown in figure 3.9. The IC will provide an output of 3V at full current (50mV input from the shunt). Therefore, the reference voltage for the ADC is chosen to be 3V, to get the most possible resolution from the current sense signal. To test this circuit, a input from -50mV to +50mV was applied and the output was observed to change from 0 to 3V, resting at 1.5V for 0V input.

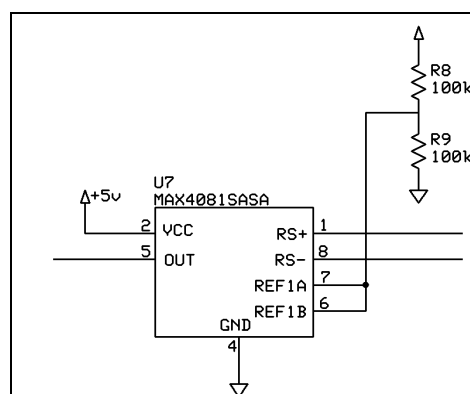
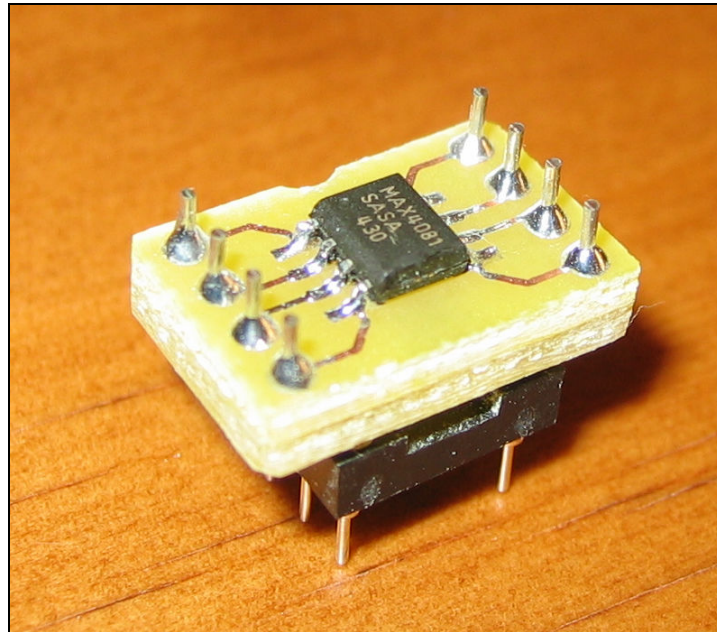


Figure 3.9 – Motor Current Sense Amplifier

Another issue facing the use of the Maxim current sense IC is that the device is only available in a SOIC-8 surface mount package. This makes prototyping difficult, as the device cannot be inserted into a breadboard. To address this problem, a small PCB was made and the surface mount device was soldered to the board, along with pins to give the chip a standard DIP pin arrangement. A photo of the adaptor PCB is shown in figure 3.10.



**Figure 3.10 – Surface Mount to DIP Adaptor PCB**

### **3.2.2 Motor speed**

To prevent the motor from over-speeding, a speed limit can be set as one of the adjustable parameters. To measure this motor speed, a sensor is placed on the motor shaft to count the revolutions. This can be done with optical or magnetic sensors. The output from these sensors will be a square wave pulse with a frequency proportional to the motor speed. This pulse can be fed directly into the microcontroller and the speed determined by measuring frequency in software, however, this takes time for the processor to count and time the pulses. Since high accuracy is not required for the speed limit, and monitoring the current limit is more critical, the microcontroller cannot afford to spend too much time measuring the speed signal.

A simple way to overcome this problem is to use the LM2917 frequency to voltage converter IC. This IC can convert a frequency input into a voltage that can be measured by the microcontrollers ADC. By taking this approach, the microcontroller does not have to count pulses or time intervals, as the external tachometer circuit achieves these tasks. The operation of the device is described as follows (National Semiconductor 2003);



The LM2917 tachometer IC is designed for a minimum external part count. The first stage of operation is a differential amplifier driving a positive feedback flip-flop circuit. The input threshold voltage is the amount of differential input voltage at which the output of this stage changes state. The differential input gives the option of setting the input switching level and still have the hysteresis for excellent noise rejection in any application.

Following the input stage is the charge pump that converts the input frequency to a dc voltage. This step requires one timing capacitor, one output resistor, and an integrating capacitor. When the input stage changes state (due to a differential voltage on the input) the timing capacitor is either charged or discharged linearly between two voltages whose difference is  $V_{CC}/2$ . Then in one half cycle of the input frequency the change in charge on the timing capacitor is equal to  $V_{CC}/2 \times C1$ . The average amount of current pumped into or out of the capacitor then is: The output circuit mirrors this current very accurately into the load resistor R1, connected to ground, such that if the pulses of current are integrated with a filter capacitor, then  $V_O = i_c \times R1$ , and the total conversion equation becomes:

$$V_O = V_{CC} \times f_{IN} \times C1 \times R1 \times K \quad \text{where } K \text{ is the gain constant (typically 1.0).}$$

The size of C2 is dependent only on the amount of ripple voltage allowable and the required response time. There are some limitations on the choice of R1 and C1 that should be considered for optimum performance. There are guidelines in the data sheet that must be adhered to for proper operation.

The LM2917 offers a built in zener for those applications where an output voltage or current must be obtained independent of supply voltage variations. The most important consideration in choosing a dropping resistor from the unregulated supply to the device is that the tachometer and op amp circuitry alone require about 3 mA at the voltage level provided by the zener. At low supply voltages there must be some current flowing in the resistor above the 3 mA circuit current to operate the regulator. In this case the supply is regulated 5V, the zener is not required and nor will it function properly as a regulator at this low supply voltage.

In this application, the optical sensor on the motor's shaft will give two pulses per revolution. For a maximum speed of 6000r/min, this equates to an input frequency of 200Hz. The circuit in figure 3.11 below is used to convert the square wave signal into a voltage suitable for input into the microcontroller's ADC. The component values are chosen to achieve a voltage of 0-3V over the expected range of motor speed.

Using the equation from the data sheet;

$$V_o = V_{cc} \times f_m \times C1 \times R1$$

$$3V = 5V \times 200Hz \times C1 \times R1$$

$$\therefore C1 = 10nF, R1 = 300k\Omega$$

$$3V = 5V \times 200Hz \times 10nF \times 300k\Omega$$

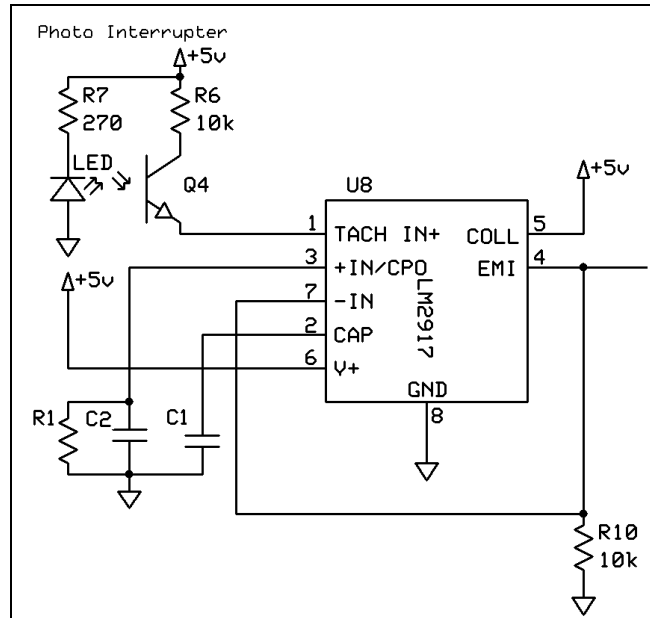


Figure 3.11 – LM2917 Tachometer Circuit

### 3.2.3 Accelerator/Decelerator pedals

The accelerator and decelerator pedals are simply potentiometers connected as voltage dividers to the 3V analog reference to give a 0-100% position reading to the microcontroller. The software uses this reading to adjust the PWM duty cycle, but leaves a dead zone at the beginning and end of the potentiometer travel to allow for mechanical issues such as throttle cable slack, etc. The potentiometers are connected as per the circuit in figure 3.12.

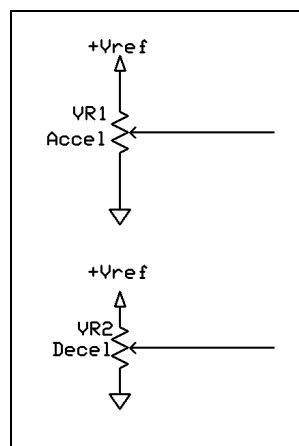
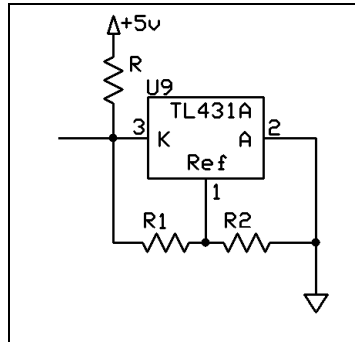


Figure 3.12 – Accelerator and Decelerator Inputs

### 3.2.4 Voltage reference

A precision voltage reference is required to ensure the ADC conversions are as accurate as possible. The microcontroller's built in reference is not accurate enough for this, as its voltage drifts with temperature and time. A common and cheap voltage reference is the TL431C programmable shunt regulator from Fairchild Semiconductor. It has an internal reference of 2.5V and has quite a low drift with temperature, 30ppm/°C. It is connected as shown in figure 3.13



**Figure 3.13 – Programmable Shunt Regulator Voltage Reference**

The output voltage, as measured at pin 3, can be calculated by;

$$V_o = \left(1 + \frac{R1}{R2}\right) V_{ref}$$

To achieve the 3V required for the reference, the following component values are required;

$$3V = \left(1 + \frac{200\Omega}{1k\Omega}\right) 2.5V$$

Therefore,  $R1 = 200\Omega$ ,  $R2 = 1k\Omega$

Slight trimming of these values may be required to achieve exactly 3V at the output. This is due to component tolerances but should only need to be done when the circuit is built. Precision voltage references that are more accurate than this circuit and do not require trimming are available, but their expense is not justified in this design.

## Section 3.3 – Gate Drive

### 3.3.1 Gate drive IC

The gate drive required to switch the high and low side MOSFETs on and off is provided by a dedicated MOSFET driver IC as discussed previously in Chapter 2. The particular chip chosen is the LM5101A high voltage gate driver from National Semiconductor. This chip is designed to drive both the high side and the low side N-Channel MOSFETs in a synchronous buck or a half bridge configuration. The floating high-side driver is capable of operating with supply voltages up to 100V. It is capable of providing currents up to 3A to drive the MOSFET gates. The outputs are independently controlled with TTL input thresholds. An integrated high voltage diode is provided to charge the high side gate drive bootstrap capacitor. A robust level shifter operates at high speed while consuming low power and providing clean level transitions from the control logic to the high side gate driver. Under-voltage lockout is provided on both the low side and the high side power rails. This device is only available in a SOIC surface mount package, so the same PCB adaptor that was used on the Maxim current sense IC was used to allow the LM5101A to be installed on a breadboard.

The IC was tested by connecting it as per the circuit diagram in the data sheets and controlling a small resistive load by applying a 5V square wave signal to the inputs. The circuit diagram is shown in Figure 3.14.

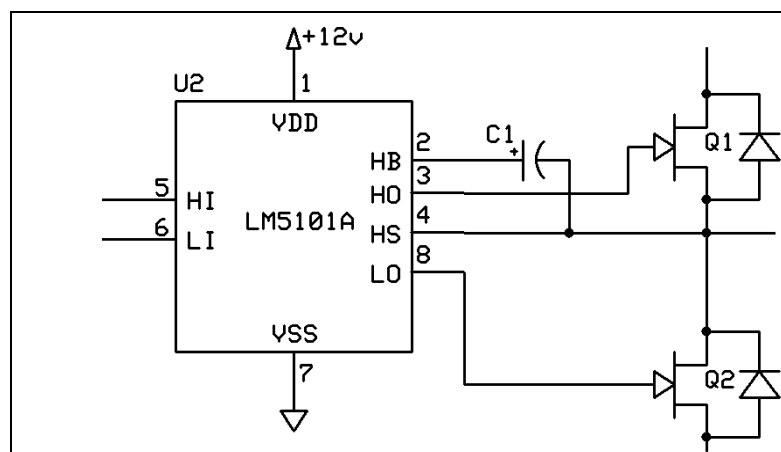
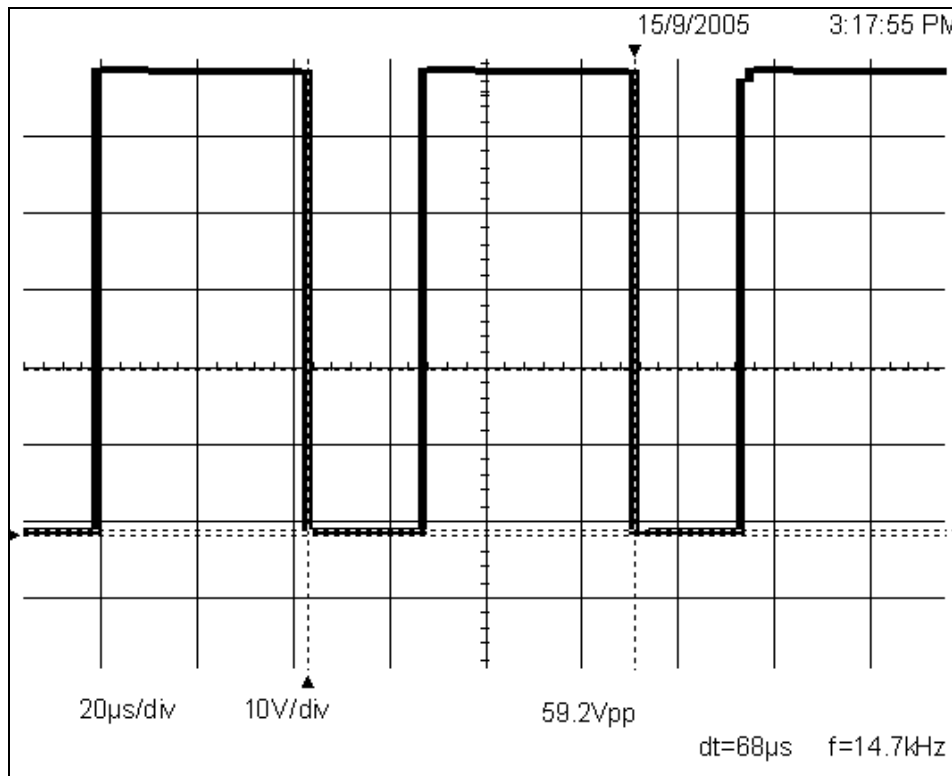


Figure 3.14 – Half Bridge MOSFET Driver Circuit

To ensure the bootstrap circuit worked properly the waveform at the high side gate driver's gate terminal was checked. This waveform is shown in Figure 3.15. The testing proved that the gate drive circuit operated as required and was suitable for use in this design. The gate drive voltages were correct, the bootstrap worked properly to provide a high side gate drive around voltage 12V higher than the its source terminal voltage.



**Figure 3.15 – Half Bridge MOSFET Driver Test Waveforms**

### 3.3.2 Protection logic

To prevent switching both the high and low side MOSFETs on at the same time, some additional logic hardware is included to direct the PWM signal to only one half of the bridge at a time. This will help prevent the occurrence of a “shoot-through” condition which would destroy the MOSFETs. The circuit, shown in figure 3.16, is constructed from two ICs – a 74HC00 2 input quad NAND gate and a 74HC08 2 input quad AND gate. It has three input signals, the PWM signal and two signals to select either the high side or low side MOSFETs.

The first gate in the circuit will only output a logic 1 if only one (or none) of the high/low side selection inputs are at a logic high. The output of this gate is fed into a second, along with the PWM signal. If there is a valid high/low side selection signal, then the output of this gate will be the PWM signal. The next pair of gates sends the inverted PWM signal to the appropriate pin of the MOSFET driver IC, depending on whether the high or low side is selected. If neither the high or low side are selected then there is no output to the gate driver IC.

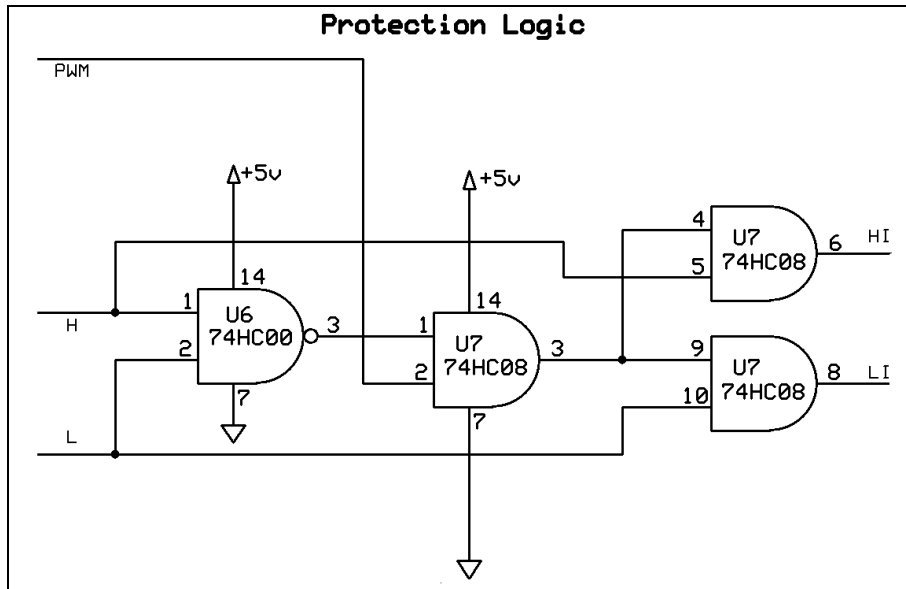


Figure 3.16 – Shoot Through Protection Logic

## Section 3.4 – Support Circuits

### 3.4.1 Power Supply

The power supply required to run the adjustable speed drive's control circuitry must be obtained from the vehicle's on-board 48V battery pack. The control circuit requires +12V to drive the MOSFET gates and +5V for the rest of the circuit. Apart from the current peaks in MOSFET gate drive current, the power consumption of the circuit should be fairly low.

The 48V lead-acid battery pack can reach voltages up to 60V when fully charged, so the regulator must be able to operate at this maximum voltage level. To step this voltage down to 12V with minimal loss, a switch-mode power supply is used. This type of regulator was chosen over a conventional linear regulator because the power dissipated in the linear regulator would be too high, up to 140 watts with a 3 amp load! The switch-mode regulator chosen is the popular LM2576 from National Semiconductor.

The LM2576 series of regulators are integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving 3A load with excellent line and load regulation. Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation and a fixed-frequency oscillator. The LM2576 series offers a high-efficiency replacement for popular three-terminal linear regulators. It substantially reduces the size of the heat sink, and in some cases no heat sink is required (National Semiconductor 1999).

To provide the 5V supply to power the microcontroller and its associated circuits, a simple LM7805 linear regulator IC is used. It is capable of providing up to 1A. Since the current required by these circuits is low, the power dissipation will be minimal. A circuit diagram of the entire power supply circuit used is illustrated in figure 3.17 below.

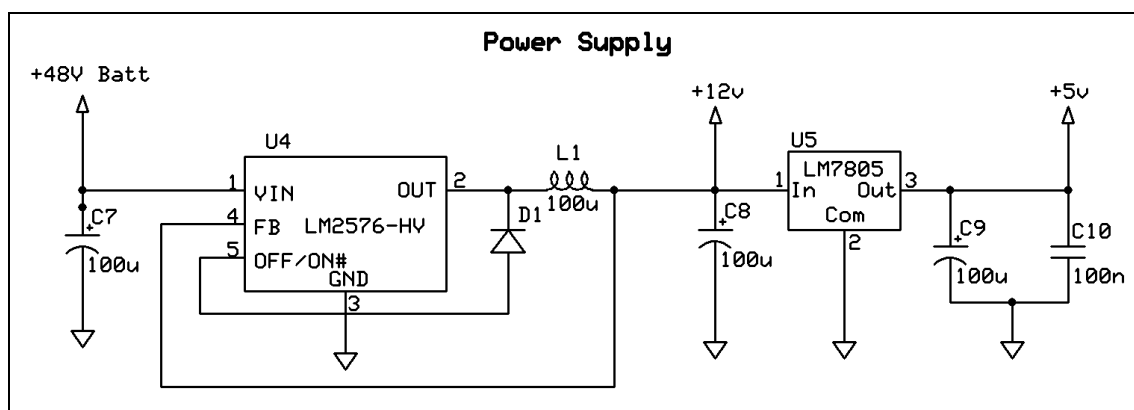


Figure 3.17 – Power Supply

### 3.4.2 Precharge circuit

The “precharge” circuit is used to charge the ripple filter capacitors prior to switching on the main power contactor. The value of the filter capacitors is in the order of several thousands of microfarads and their purpose is to smooth the current drawn from the batteries and cancel the effect of inductance in the power cables from the batteries. If the capacitance was not there, the inductance of the battery cables will cause undesirable effect on the power stage. In addition, battery current would be drawn in large pulses and this increases stress on the batteries. As previously mentioned, the capacitors must be pre-charged before switching on the main contactor to prevent damaging it from the large inrush current caused by the empty capacitors.

To overcome this problem, the capacitors are charged via a current limiting resistor prior to switching on the main contactor. This is achieved by the microcontroller via a transistor and a voltage input to its ADC to measure capacitor voltage. The microcontroller switches on a transistor which lets current flow into the capacitors via a limiting resistor. The voltage across the capacitors is measured and once it begins to approach the battery voltage, the main contactor is switched on.

This circuit can also be used as part of a fault detection system, if the capacitors fail to charge then there must be a fault in the power stage and the drive should not be enabled. Faulty capacitors or a short circuited power stage will prevent the capacitors from charging. In addition, if the capacitors appear to charge too quickly (i.e. faster than the expected RC time constant), the capacitors may be faulty.



# Chapter 4

## 4.0 Control System Implementation and Testing

The control system for the adjustable speed drive is handled by the microcontroller, hence this chapter is concerned with how the microcontroller is used to facilitate driving the power stage in accordance to the driver's input and the system status. Leading on from Chapter 3, this chapter describes how all the elements in the previous chapter come together to form the system as a whole, along with an explanation of the software programmed into the microcontroller and PC. The results of tests performed on the control system are also presented.

### Section 4.1 – Overall Description

Figure 4.1 illustrates a block diagram of the whole system, showing how each of the individual circuits presented in Chapter 3 link together. All the inputs to the system are converted to digital form by the microcontroller and are processed by the control system software, which in turn provides the outputs to the power stage. The inputs from the driver are an accelerator and brake pedal, just like any normal vehicle, as well as a switch to select forward or reverse. The goal is to make the control system operate the motor so that it operates in an intuitive way and has the same response to drivers' inputs just as any other ordinary vehicle would.

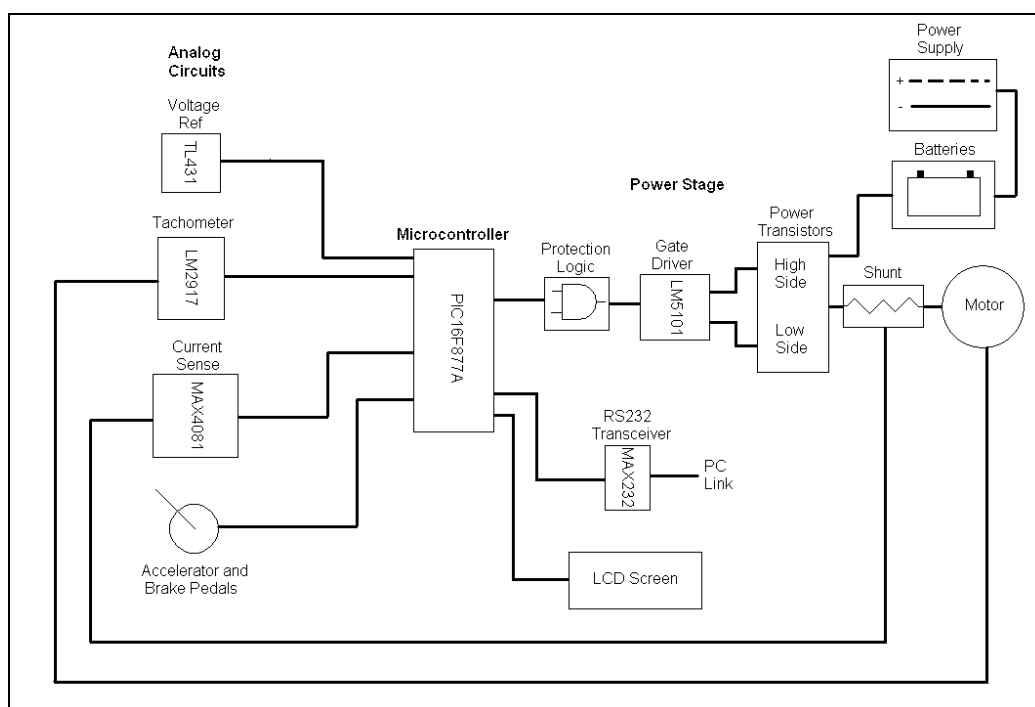


Figure 4.1 – Functional Block Diagram

A relatively simple control system is required to achieve this goal, as the human driver forms part of the feedback loop. There is no need for PI or PID controllers, the system must simply respond as per the driver's instantaneous commands. If the driver wants the vehicle to speed up, they simply press down more on the accelerator. When they have reached their desired speed, they simply ease off the accelerator until that speed is maintained. This "human feedback loop" is the same principle used in any ordinary automobile. There is some feedback to the microcontroller in the form of current, voltage, and speed, but this is only used to protect the circuit and ensure operation is only within the user-specified parameters.

The control system, in simple terms, reads the accelerator or brake pedal position and receives a value from 0-100%. It then sets the PWM duty cycle accordingly, from 0-100%. If the system senses that the motor current or speed is too high, it will automatically cut back the duty cycle to protect the electronics and the motor.

Of course, the practical implementation of this system is somewhat more complex. There are other issues to consider. For example, what happens if the driver selects reverse when the vehicle is travelling forwards? What happens if the driver tries to do regenerative braking and the batteries are already charged? What happens if the electronics short out or the accelerator fails and the motor gets stuck full on? These issues are addressed by the control system software and the circuit design.

One important safety aspect to consider is the braking system. The motor control can provide regenerative braking, but since the braking energy must go into the batteries it cannot operate if the batteries are fully charged. This means that sometimes the drive will not be able to provide any braking force. To address this issue, the vehicle must have a mechanical brake that is sufficient to do all the braking on its own. The electrical and mechanical brake would still be controlled by the same pedal, the first portion of brake pedal travel will be regenerative braking only, and then if the pedal is pressed further the mechanical brake will come on. Using this method the driver should not notice much of a change in braking performance even if regeneration cannot take place as they will instinctively push down more on the pedal until the desired amount of braking is achieved.

Another important issue is safe start up and shut down of the power stage to ensure the vehicle doesn't move unexpectedly due to a fault condition. This issue is addressed in the microcontroller's software. When the system is powered up, the power stage and main contactor is switched off. The software performs "self-tests" to ensure that no pedals are being pressed and that the power stage circuit pre-charges properly before turning on.

These measures will detect if a potentiometer is faulty or a pedal is accidentally held down at switch on, as well as ensuring the high side of the power stage isn't shorted before turning on the power. This should prevent the vehicle from moving suddenly or unexpectedly when the power is turned on, either due to a fault or inadvertent pressing of the pedals.

Additional circuitry could be added to detect faults with the accelerator or brake potentiometer wiring during operation (i.e. after the start up "self-test"). There is the remote possibility that the ground wire of the potentiometer may become severed or faulty, this will cause the microcontroller to read the input as a WOT (pedal all the way on) and go to 100% duty cycle. However, this protection is not included as the system should be shutdown by the speed limiting or operator intervention.

## Section 4.2 – Microcontroller Implementation

As discussed in Chapter 3, the Microchip PIC16F877A microcontroller was chosen for use in this design. This section discusses how the microcontroller's features are used in this design, including the way various analog inputs and digital outputs are interfaced to it. Figure 4.2 represents the inputs and outputs of the microcontroller.

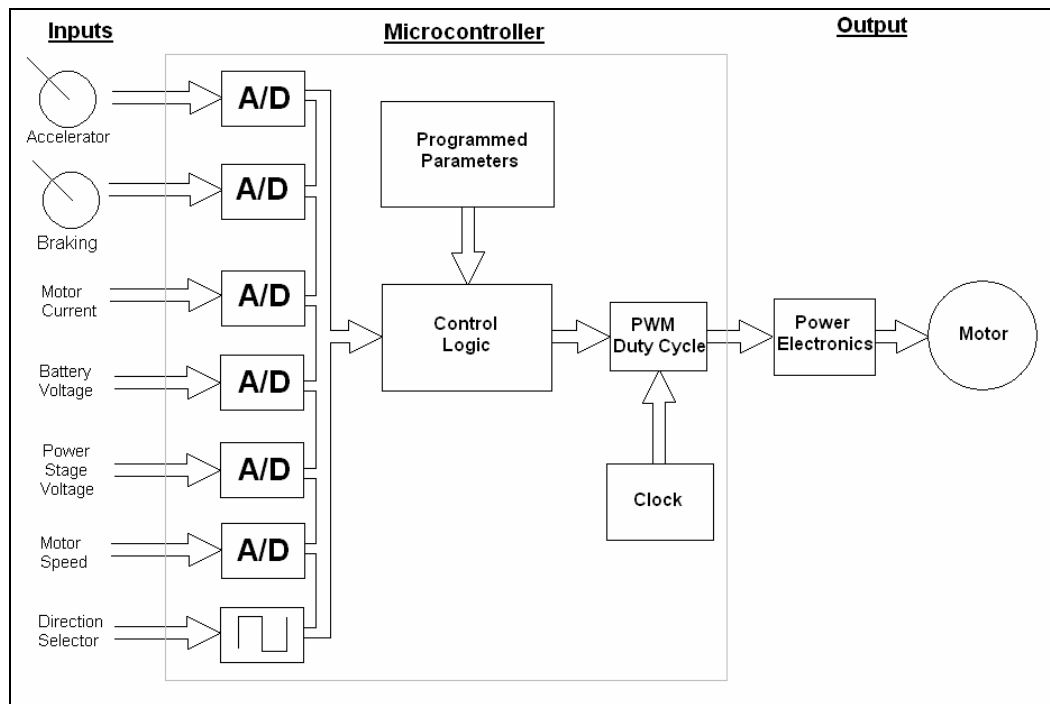


Figure 4.2 – Microcontroller Inputs and Outputs

As depicted in Figure 4.2, the majority of the inputs are analog and are measured by the microcontroller's built-in analog to digital converter. The only digital input is a forward/reverse selection switch. The analog inputs are processed by the circuits described in Chapter 3 before they are applied to the microcontroller's ADC in the form of a 0-3V signal. These inputs are then read by the control system software so it can control the output accordingly.

The output from the microcontroller to the power stage is in digital form. There are six outputs from the microcontroller to the power stage; the PWM signal, a high side select signal, a low side select signal, pre-charge enable, main contactor enable signal, and a forward/reverse contactor control signal. The PWM and high & low side select signals are connected to the protection logic circuit. This circuit ensures that only the PWM signal is applied to either the high or low side at any particular time. The PWM output to the power stage can be switched off by setting both high & low side select signals to a logic 0 or a logic 1.

The main contactor and forward/reverse contactor control signals are fed to transistors which in turn switch the contactors on and off as required. The pre-charge enable line also switches a transistor to allow pre-charging of the power stage filter capacitors through a resistor prior to switching on the main contactor.

In addition to the power stage outputs, there is also six digital output lines to the LCD display and two lines from the microcontroller's USART used as transmit and receive lines. The two lines from the USART are for the RS-232 serial link, used when setting the drive parameters. Further description of the microcontroller's implementation in an actual circuit is given in Section 4.5.

Problems can arise when mixing sensitive microcontroller circuits with high-energy power stages. Protecting the microcontroller by using good design practices is very important as it manages all of the design's functionality. There are many problem areas facing a motor control design, some of which include:

- Intermittent connections (e.g. loose wires or connectors)
- Voltage spikes and surges
- Reversal of power connections (user wiring error)
- Power stage transistor short
- Microcontroller error due to undiscovered software bugs
- Electrostatic discharge damage to active components
- Motor failure (eg short circuit, locked rotor, etc)
- Other unknown failure modes due to unexpected faults

Microcontroller based designs used in motor controls are more prone to damage than those used in office equipment, for instance. This is because the operating environment is much harsher, and the microcontroller must operate in the presence of large voltage and current transients.



## Section 4.3 – EMC and EMI Considerations

A considerable amount of attention should be paid to electrical noise management when designing a motion control system. The circuit must be robust in the presence of high levels of noise, since the power stage will unavoidably produce significant amounts of noise. Attempts should also be made to help prevent as much noise radiating from the circuit as possible so that it does not interfere with itself or other electronics that might be operating nearby. This section will focus on the noise in the control circuit; Section 5.3 of Chapter 5 concerns the EMI and EMC issues of the power stage.

### 4.3.1 Noise and Noise Immunity

The control circuit design will require filtering and noise immunity if it is to operate a high power output stage, extensive testing will determine the actual extent of the measures required to combat noise. It should be noted that while the circuit illustrated in this document has some noise filtering included, it may not have sufficient noise immunity to ensure proper operation in a very high noise environment. This section will discuss the various methods that can be implemented to enhance noise immunity, but they cannot all be implemented due to time and cost constraints.

Figure 4.5 depicts the major noise problem areas in a microprocessor controlled motor drive system. As can be seen, there are many sources of EMI, as well as many opportunities for the noise to enter parts of the circuit where it can cause problems. Such areas include analog inputs, power supply lines, I/O lines, etc.

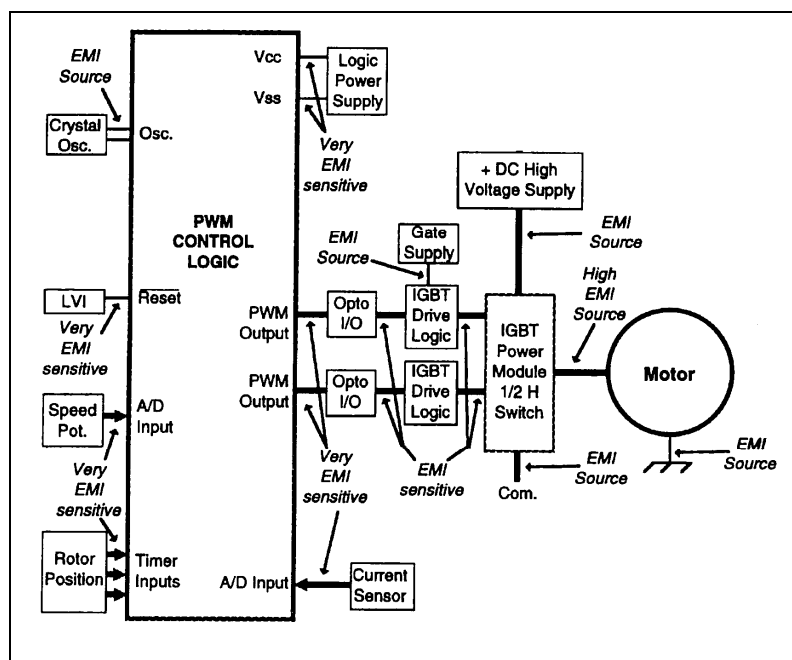


Figure 4.5 – Electrical Noise Problem Areas in Motor Drive Systems (Valentine 1998)

If noise spikes managed to find their way to the microcontroller, they can induce erratic operation or even cause damage to the microcontroller. The software may crash or internal registers could be corrupted, this could lead to unpredictable operation which could possibly have disastrous results such as power stage damage or loss of vehicle control. According to Valentine 1998, one quick test recommended by EMI experts is to operate a high powered mobile phone in close proximity to the circuit. If the circuit operates erratically, serious EMI testing must be undertaken. Another simple test to determine EMI tolerance is to operate the circuit near an automotive ignition system.

### 4.3.2 Layout

Printed circuit board layout plays a large role in determining the EMI and EMC of a design. General rules of thumb to minimise EMI during PCB design are listed below:

- Use the shortest and widest traces possible, a “3-1” layout rule has been suggested by EMI experts. That is, the trace width should not be less than one-third of its length.
- Route clock lines or high-speed lines near  $V_{cc}$  or  $V_{ss}$ .
- Analog lines should be routed carefully, the ADC reference should not be mixed in with  $V_{cc}$  or  $V_{ss}$  or other traces carrying pulsed currents, and separate grounds should be used for analog and digital circuits.
- Proper and regular placement of filter capacitors. Decoupling or bypass capacitors should be placed as close as possible to the source of high-frequency noise.
- Follow data sheet or application-note recommended PCB layout rules if possible. Microcontroller clock oscillator circuits are a potential source of EMI and manufacturers usually give a recommended oscillator layout in the data sheet.
- Minimise loop areas, loops act as antennas that may radiate or receive noise
- Cancel fields by running equal currents that flow in opposite directions as close as possible to each other
- T connections and  $90^\circ$  angles should be avoided on traces that carry high speed signals as these represent a discontinuity to the wavefront, resulting in unwanted reflections.
- Connect circuit grounds to power grounds at only one point. This reduces the effect of transient voltage drops along the power grounds due to high  $di/dt$  occurrences.



### 4.3.3 Circuit Design

The issue of circuit design was discussed in the previous section; however there are a few further points to consider.

Sufficient filtering on microcontroller input lines and power supply rails is critical, as all cables, traces, and device leads exhibit some inductance. If there is insufficient filtering then the  $V_{cc}$  and commons will become noisy and possibly lead to undesirable operation of the circuit. Low pass filters on the ADC input lines, with a corner frequency around 10 times lower than the PWM frequency are recommended by some designers.

If the circuit is to operate at high voltages ( $>150V$ ), optocouplers or some other form of electrical isolation should be used to separate the low voltage microcontroller circuits from the high voltage inputs and outputs. This will help reduce noise transmission and prevent damage to the sensitive microcontroller circuit. Slowing down switching edges, if possible, will also reduce the amount of noise created due to the reduction in high frequency harmonics present in the switching signals.

## Section 4.4 – Software

This section describes how the control system is implemented in software and how the PC program is used to transfer operating parameter to the microcontroller's EEPROM.

### 4.4.1 Microcontroller Firmware

The software for the microcontroller (known as “firmware”) is written in the BASIC programming language. As mentioned previously in Chapter 3, a software package called Mikrobasic was used. It provides a full development environment and includes a compiler for the PIC16F877A. The development of the firmware involved many stages of testing and debugging. As sections of the code were written, they were programmed into the microcontroller and tested to ensure they worked properly. A fully commented program listing of the final version of the firmware can be found in Appendix E. This sub-section explains the operation of the firmware step by step.

When the motor controller is first powered up, the LCD is initialised and a welcome message is displayed on the screen. The microcontroller then proceeds to initialise the other ports by setting their data direction registers to the appropriate values, turning off all outputs. It also sets up the PWM module for an output frequency of 15 kHz and configures the ADC. Once this is complete, it then reads the drive parameters from the EEPROM and checks their validity. If they appear to be invalid, the system enters the “programming mode” to allow the drive parameters to be sent via a connection to a PC. The system will also enter programming mode at this point if the “programming mode selection” switch is on.

In programming mode, the USART is initialised for 2400 baud and the LCD announces that the system is in programming mode and is ready to be connected to a PC. It then waits for data input from the PC. Once the connection is established, the microcontroller sends out the current drive parameters via the USART. It then waits for the new parameters to be sent from the PC. Once the user has entered the new parameters into the PC and sends them, the microcontroller reads these values from the UART and stores them into the EEPROM. It then shuts down the controller so that it can be restarted to load the new operating parameters.

If there was no requirement to enter the programming mode, the microcontroller continues its start-up process. It checks to ensure that the battery voltage is within the programmed limits and shuts down the system with an appropriate error message displayed if there is a problem. Next, the microcontroller checks to ensure that the brake and accelerator pedals are reading zero. If they are not, the LCD displays the error condition and again shuts down the controller. This is to prevent the vehicle moving unexpectedly.

If all is well thus far, the microcontroller begins the pre-charge sequence. This involves switching on the pre-charge transistor to allow the power stage's filter capacitors to charge via a resistor. The voltage of the capacitors is monitored via the ADC, and if it does not rise to within approximately 2V of the battery voltage in ten seconds the system will display an error message and shut down. If the capacitors charged successfully then the main contactor is switched on and the reversing contactors are set to achieve the motoring direction selected by the forward/reverse switch. The LCD display indicates whether the drive is in forward or reverse.

The firmware then enters into an endless loop which performs the following steps:

1. The watchdog timer is reset to indicate that the software has not crashed and the microcontroller need not be reset
2. The battery voltage is read and checked. If the drive is operating in the regeneration mode and the battery voltage has exceeded 2.5V per cell (the maximum safe voltage according to Linden 1995) then the PWM signal will be set to zero and the cycle skipped to prevent damaging the batteries from overcharge. If the drive is not in regeneration mode and battery voltage is otherwise outside the preset limits, then the drive will shut down and display the fault condition on the LCD.
3. The motor speed is checked to ensure it has not exceeded the preset limit for the current mode of operation (there are separate speed limits for forward and reverse). If it has, then the PWM signal is switched off and the cycle is skipped.

4. The direction switch is checked and if it has changed state then the reversing contactors are controlled accordingly. The motor control must switch from forward to reverse direction in a safe manner. This means ensuring the motor has come to a stop before switching the reversing contactors. Common sense would say that the driver would only change into reverse if the vehicle is stopped, but the switch could accidentally be pressed while travelling at speed. When the forward/reverse switch changes state, the software shuts off the PWM output and waits for the motor speed to drop to almost zero. Once the motor has stopped, the field coils are then reversed with the contactors.
  
5. The accelerator and brake pedal positions are then read. If the brake pedal is pressed, regeneration mode is entered. If the accelerator pedal is pressed, motoring mode is entered. The brake pedal overrides the accelerator pedal. Once the pedals have been read, the motor current is checked to ensure it is below the limit for the set current quadrant. If it is not, then the PWM signal is switched off and the cycle is skipped. Otherwise, either the high or low side is selected as required, and the PWM duty cycle is set according to the position of the pedal (0-100%). The software then enters another loop that monitors the current to see if it has exceeded the value set by the drive parameters. There are individual current limits for forward, reverse, and regeneration. If the current limit is exceeded during the PWM on-time then it is shut off until the next cycle.

This sequence of events repeats infinitely until the power to the system is switched off. The firmware implements all of the features listed in Chapter 1. The effectiveness of the control firmware is tested in Section 4.6 of this chapter.

#### 4.4.2 PC Software

The software written to allow setup of the drive parameters and programming of the microcontroller's EEPROM via a RS232 connection was written in Visual Basic 6 and utilises an ActiveX library obtained on the internet from National Control Devices. The library allows byte level communication with the PC's RS232 port. A full listing of the program code can be found in Appendix F. The purpose of the software is threefold:

1. To provide a user interface for the operator to easily program the drive parameters
2. To convert the user's desired parameters into values that can be compared with the microcontroller's ADC, and
3. To provide a communication link between the microcontroller and the PC to program the EEPROM.

The user interface which was designed is pictured in Figure 4.6. It provides an easy way for an operator to enter the desired settings in an intuitive manner.

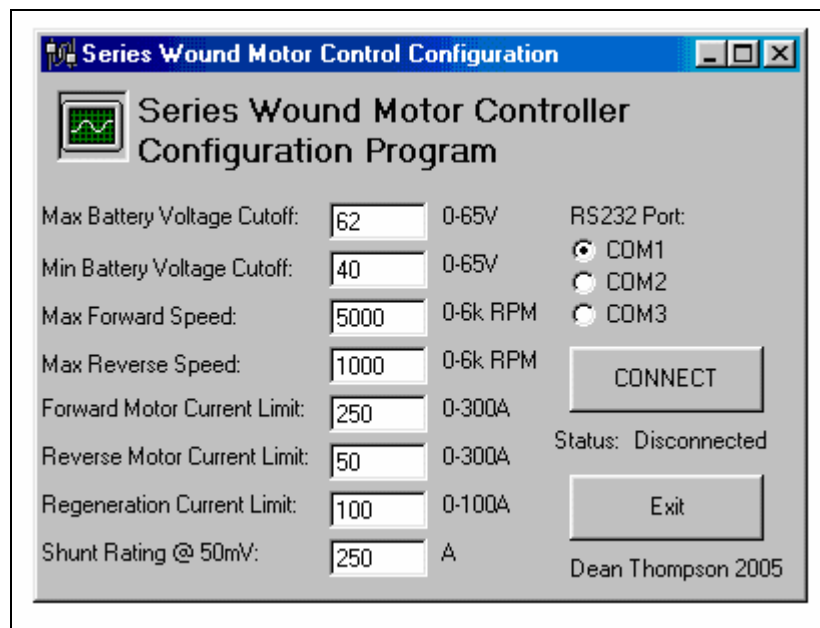


Figure 4.6 – Screenshot of PC Configuration Software

When the PC is connected to the microcontroller and it is in programming mode, clicking the “Connect” button will load the currently stored parameters from the microcontroller's EEPROM to the PC where they are displayed on the screen. The values can then be edited, and then sent back to the EEPROM by clicking the “Connect” button again.

The drive parameters are stored in the EEPROM in the form of values that are read from the ADC, i.e. a 10 bit number (0-1023). This means that the values entered by the users, which are in volts, amps, and r/min, must be translated into their equivalent ADC reading. The ADC input range is 0-3V, it measures this voltage and assigns it one of 1024 levels. To convert the drive parameters to their corresponding ADC readings the values need to be multiplied by a scaling factor. This factor depends on the circuit design, for example the tachometer circuit gives an output of 0-3V over the range of 0-6000r/min. Thus to convert a value in revolutions per minute to its corresponding ADC reading it must be multiplied by the maximum ADC reading divided by the maximum r/min input. A list of all the parameters, their range, conversion factor and EEPROM location is presented in table 4.1.

Location (0-255)	Parameter	Range	Conversion Factor	Storage Format
0-1	Max battery voltage	0-65 (V)	$\frac{1023}{65} = 15.738$	16 bit number representing 10bit ADC output
2-3	Min battery voltage	0-65 (V)	$\frac{1023}{65} = 15.738$	16 bit number representing 10bit ADC output
4-5	Max forward speed	0-6000 (R/min)	$\frac{1023}{6000} = 0.1705$	16 bit number representing 10bit ADC output
6-7	Max reverse speed	0-6000 (R/min)	$\frac{1023}{6000} = 0.1705$	16 bit number representing 10bit ADC output
8-9	Max forward motor current	0-Shunt rating (A)	$\left(\frac{511}{Shunt\ Rating}\right) + 512$	16 bit number representing 10bit ADC output
10-11	Max reverse motor current	0-Shunt rating (A)	$\left(\frac{511}{Shunt\ Rating}\right) + 512$	16 bit number representing 10bit ADC output
12-13	Max regen current	0-Shunt rating (A)	$512 - \left(\frac{511}{Shunt\ Rating}\right)$	16 bit number representing 10bit ADC output
14-15	Shunt current at 50mV	Shunt rating (A)	N/A	16 bit number stating amps in an integer format

**Table 4.1 – Drive Parameters**

The ADC readings for the current sense input are slightly more complex to calculate. This is because the current sense is bi-directional. The Maxim current sense amplifier chip provides an output of 1.5V to the ADC when the current is zero, 0V when the current is at its maximum negative value, and 3V when it is at its maximum positive value. The maximum measurable current level is defined by the shunt rating. This means there are only 512 levels for each direction of current flow, giving a total of 1024.

The PC software was tested and successfully works to read and program the EEPROM with the correct values.

## Section 4.5 – Overall Circuit Operation

This section presents the entire control circuit and explains various aspects of the design. The complete circuit design incorporates the various individual circuits that were designed and tested in Chapter 3. A schematic detailing the entire circuit design can be found in Appendix D. The power stage circuit will be discussed further in Chapter 5.

A full listing of all the microcontroller’s I/O lines and their assigned functions is given in table 4.2. These pin assignments were chosen according the specifications in the microcontroller’s data sheets. A feature list and pinout of the microcontroller is given in the datasheet extract presented in Appendix B. Most of the available pins are used in this design, a smaller microcontroller would reduce cost but not provide sufficient I/O lines to achieve all the desired functions.

Port	Direction	Function
RA0	Analog In	Accelerator
RA1	Analog In	Brake
RA2	Analog In	Motor current
RA3	Vref	Analog voltage reference
RA4	-	-
RA5	Analog In	Battery voltage
RB0	-	-
RB1	Input	Program mode enable
RB2	Input	Forward/Reverse Selection
RB3	Input	Low Voltage Programming Select
RB4	-	-
RB5	-	-
RB6	Input	Programming clock signal
RB7	Input	Programming data signal
RC0	Output	Main contactor enable
RC1	Output	Pre-charge enable
RC2	Output	PWM signal
RC3	Output	High side select
RC4	Output	Low side select
RC5	Output	Reversing contactors
RC6	Output	USART TX
RC7	Input	USART RX
RD0	-	-
RD1	-	-
RD2	Output	LCD RS
RD3	Output	LCD E
RD4	Output	LCD D4
RD5	Output	LCD D5
RD6	Output	LCD D6
RD7	Output	LCD D7
RE0	Analog In	Motor speed
RE1	Analog In	Power stage check voltage
RE2	-	-

**Table 4.2 – Microcontroller Pin Assignment**

While the design of most sections of the circuit has been discussed previously in Chapter 3, there are some additional points to note regarding the design of the circuit as a whole.

The control circuit power supply contains some refinements in comparison to the one presented in the previous chapter. There is a 4A fuse on the input and a diode to protect against reverse polarity connection of the power supply from the batteries. In addition, there are also zener diode clamps on the 48V input, the 12V output and the 5V output. If the voltage goes above the zener voltage at one of these points the diode will begin to conduct and the current will go high, blowing the fuse and shutting off the circuit. This should help to protect the circuit against over-voltage conditions. Also, the microcontroller has a built in low-voltage cut-out that will automatically shut it down if its supply voltage is too low to guarantee proper operation.

The analog input lines are fitted with low pass filters with a corner frequency around 10 times lower than the PWM frequency. They are also fitted with 3.3V zeners to clamp the voltage if it exceeds the analog  $V_{ref}$  of 3V. The digital input lines from the forward/reverse switch and programming switch are also fitted with RC filter circuits.

Resistor dividers are used to measure the power stage voltage and battery voltage via the ADC. They reduce the voltage so that an input of 0-65V will provide a 0-3V output to the ADC. Transistor buffers are used to drive the contactors and pre-charge circuit from the microcontroller's TTL data lines. Diodes are fitted across the coils of the contactors to prevent damage to their driver transistors due to inductive kickback at turnoff.

The microcontroller is connected as per data sheet recommendations. A pin header is provided for connection to the ICSP firmware programmer. The device's clock is generated from a 20MHz crystal connected with compensating capacitors.

A 9-pin D connector with a "null-modem" connection is provided for connecting the circuit to a PC's RS232 port for downloading drive parameters with the software presented in the previous section.



## Section 4.6 – Control System Testing

Although it was impractical to perform a test of the final system due to the lack of time and resources for PCB layout and construction, as well as full power stage construction, the control system could still be tested. The main elements of the control system were constructed on a breadboard. These included the microcontroller and the protection logic, as well as the LCD display and RS232 interface circuitry. Five potentiometers were used to provide the analog inputs to the circuit; they were motor speed, motor current, battery voltage, brake pedal, and accelerator pedal. There is also a toggle switch to select between forward or reverse motoring modes. The test circuit is pictured in Figure 4.7 with each of the inputs labelled.

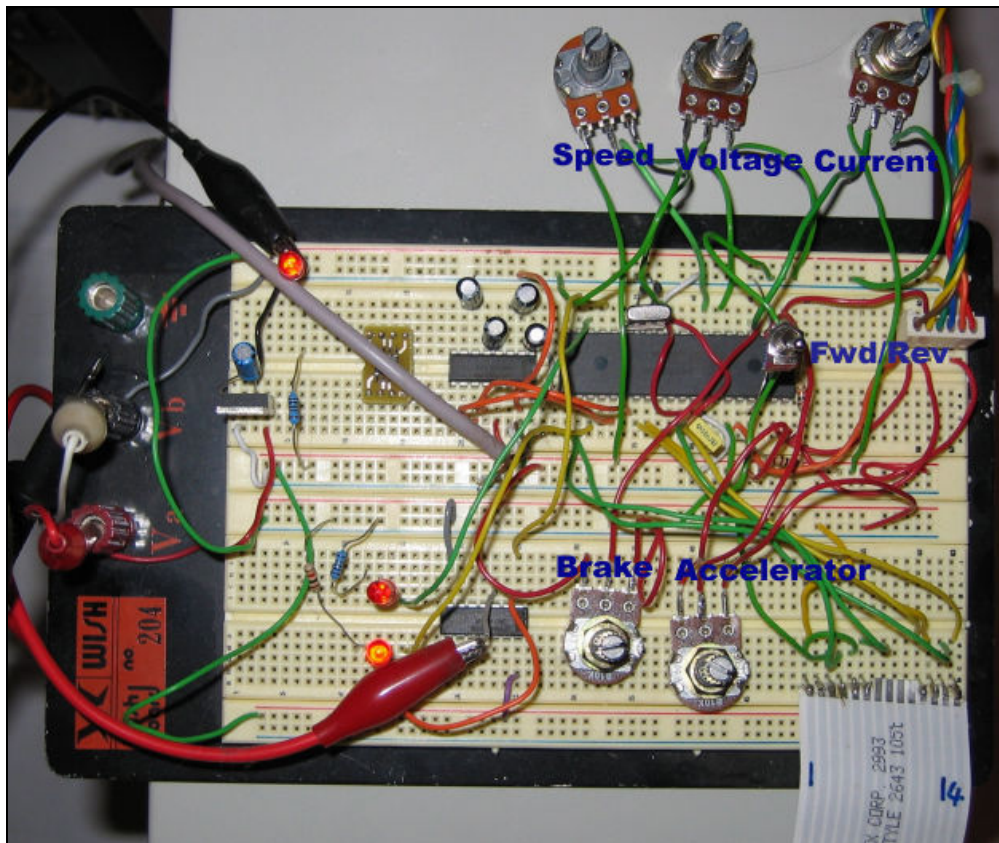


Figure 4.7 – Testing of the Control System

Testing was hampered by bugs in the compiler and PWM library, and the 2k program size limit of the demonstration software, however these issues were overcome and all functions of the control system firmware were tested successfully. To test system response times to various limit situations, a step input was applied to the input while the PWM was operating. The over current, over voltage and over speed conditions were tested. The remainder of this section describes each test that was carried out and assesses the results.

#### **4.6.1 EEPROM Parameter Storage**

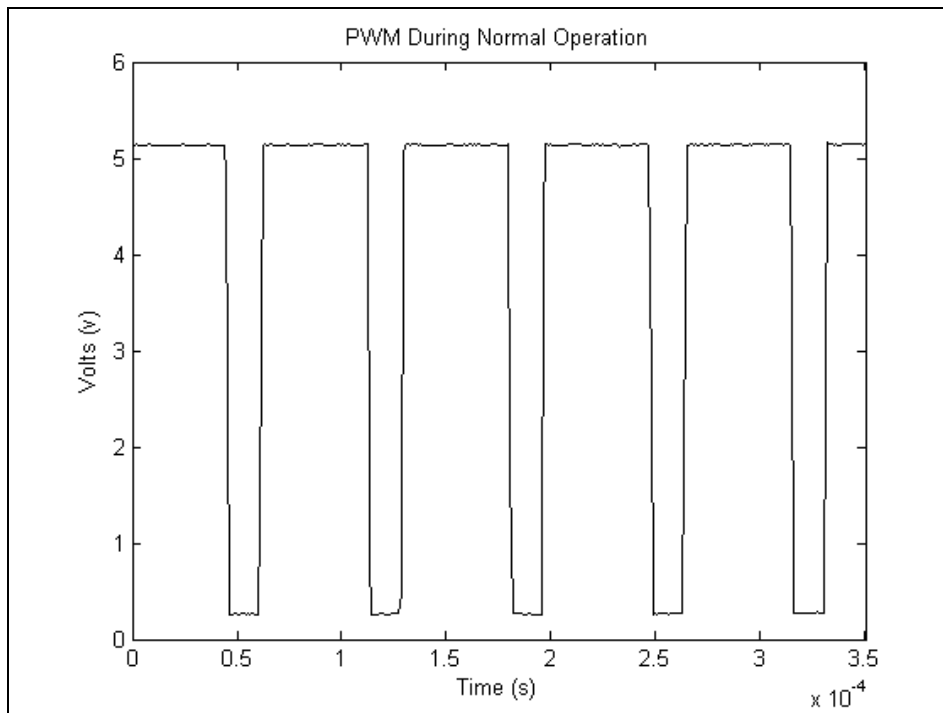
To test for correct operation of the drive parameter storage and retrieval portion of the software, a number of tests were carried out. Firstly, a set of parameters was manually programmed into the EEPROM using the development software's device programmer. The microcontroller was able to successfully read this information and use it in the motor control software. The next step was to use the Visual Basic PC software written specifically to program the EEPROM with new parameters. After some debugging of both the PC and microcontroller software, the PC was able to read the current parameters via the RS232 link and then download new settings into the EEPROM. This proved that the software in the microcontroller and PC was working properly and converting the user inputs into correct values for the microcontroller to compare with the values read by the ADC.

#### **4.6.2 Contactor Control and Pre-charge Outputs**

One important safety feature of the design is the main contactor and pre-charge circuit. When the motor control is powered up, it performs some self tests to ensure that it is safe to switch on the power stage. The control should not enable the main contactor if the battery voltage is too high or too low, if the accelerator or brake pedals are pressed, or if the pre-charge fails to charge the capacitors in the power stage. Using the potentiometers, fault conditions were simulated to ensure the software executed a safe power-up of the motor control. The system worked as expected, the control system would shut down and display an error message on the LCD screen if any parameters were outside the limits programmed in the EEPROM or if the pre-charge timed out by taking too long to charge up. The outputs of the microcontroller which control the main contactor and pre-charge transistor were also monitored to ensure that they switch on and off at the correct moments.

### 4.6.3 PWM Output

This simple test was used to ensure that the PWM signal output was correct and varied in accordance with the position of the brake or accelerator positions when no preset current, speed or voltage limits were exceeded. Figure 4.8 shows the 15kHz PWM output from the microcontroller's CCP pin as captured by the oscilloscope.



**Figure 4.8 – PWM Output During Normal Operation**

### 4.6.4 Shoot Through Protection Logic

Testing of the shoot through protection logic was undertaken to ensure that only the low side received the PWM signal during regeneration, and only the high side during motoring. Using the oscilloscope, it was found that at no time were both the high and low sides signalled to be switched on at once. Varying the accelerator and brake potentiometers proved that the circuit did indeed apply PWM to the correct side of the bridge, and that any movement of the brake pedal overrides the position of the accelerator pedal and changes the drive to regenerative braking mode.

#### **4.6.5 Forward/Reverse Switching**

To ensure the reversing contactors are only switched when the PWM signal is off and the motor has stopped spinning, the speed potentiometer was first set to indicate to the microcontroller that the motor was spinning at around 3000r/min. The selection switch was then changed from forward to reverse. The PWM signal was shut off by the microcontroller, and the LCD displayed “Spinning down”. The speed potentiometer was then turned down to zero, and the microcontroller then switched on the reversing contactors and changed the LCD display to “Reverse!” The same process was repeated, but this time the switch was changed from reverse to forward. Again, the circuit operated as expected and no bugs were discovered.

#### **4.6.6 Limiting and Protection**

There are three types of limiting in the control system; speed, voltage, and current limiting. To test that the limiting was responding correctly, the EEPROM was loaded with a typical set of operating parameters and the accelerator potentiometer set to begin the PWM output. The other potentiometers were then used to simulate situations that were outside operating limits.

The battery voltage potentiometer was varied to give an under-voltage and an over-voltage condition. In both cases the power stage was shut down, the main contactor switched off, and the LCD displayed the fault condition. Next the speed potentiometer was set to an over-speed condition. In this case, the PWM output was disabled until the motor speed was reduced below the speed limit. The software also used the correct speed limits depending on whether forward or reverse was selected.

The current limiting proved a little more difficult to get working, as its response time needs to be as fast as possible to shut off the PWM signal if the motor current exceeds a preset limit. It took some time to iron out the bugs in this part of the software but eventually it worked as required. There are three different current limits, depending on what quadrant the drive is operating in. The correct current limit was used by the software in all cases.

#### 4.6.7 Response Times to Limiting Conditions

Testing the control system's response time to the various limiting conditions required the use of an oscilloscope that could capture and hold a single event. One channel of the oscilloscope was connected to the PWM output of the protection logic (where the gate drive would be connected) and the other connected to the microcontroller's ADC input corresponding to the limit under test. The oscilloscope was set to capture a single shot event, triggered by the occurrence of the limit input. The captured data was stored to disk and plotted using Matlab.

The most critical limit is the current limit; the control system needs to respond quickly enough to switch off the PWM signal so that the motor current does not continue to rise. To test the response time, the PWM and current signals were monitored with the oscilloscope. A step input was applied to the current sense input to indicate an over-current condition. The oscilloscope was set to be triggered by the step input. The waveform captured is shown below in Figure 4.9, the blue waveform is the PWM signal and the red waveform is the step input.

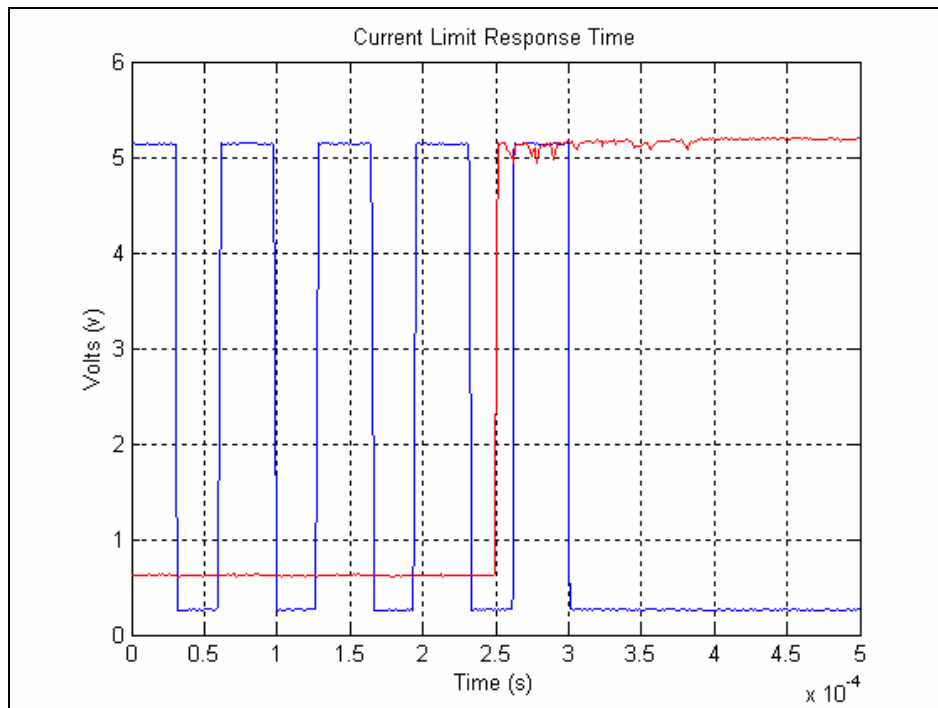


Figure 4.9 – Current Limit Response Time

The testing showed there was a 50µs delay between when the current sense input went high and the PWM signal was turned off. This delay is due to a number of factors, mainly due to the use of the development software's built in ADC read library which is probably not optimised for speed. The typical acquisition and conversion time of the ADS is 21µs, the extra delay in response time is due to the software.

The response time could possibly be improved by re-writing the software in speed optimised assembly language, however this would be quite difficult and time consuming. A faster ADC chip could also be used to speed up the response time. Also note that if the microcontroller was actually sensing motor current it would be ramping up and down as a triangle waveform as shown by the "inductor current" waveform illustrated in figure 2.8 (Chapter 2).

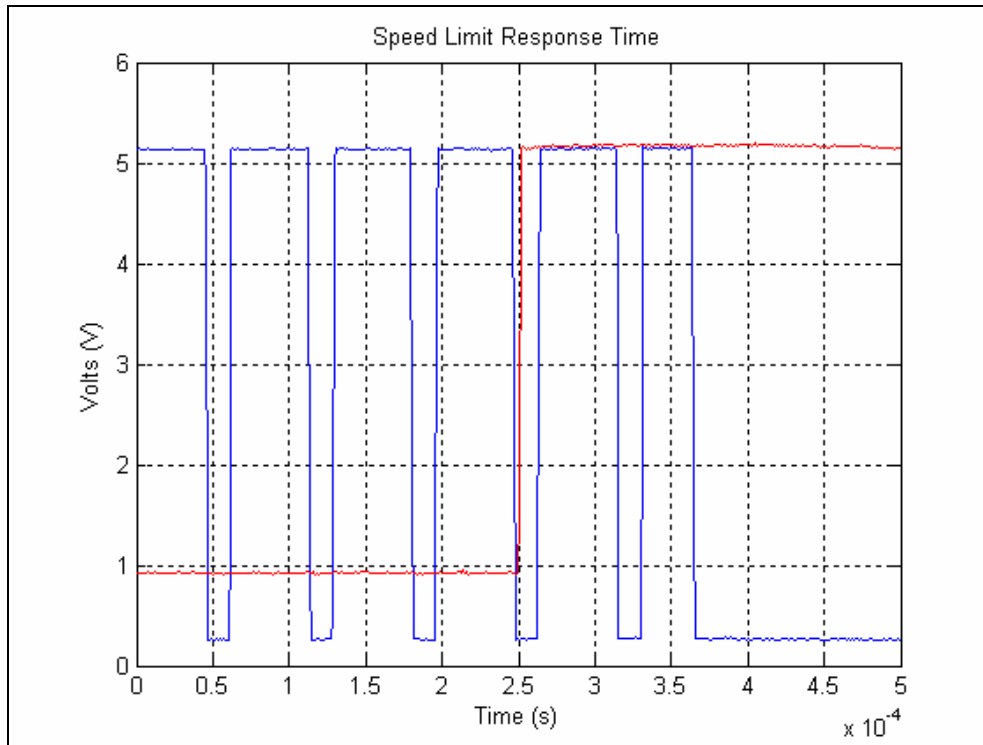
Assuming a battery voltage of 50V and motor inductance of 500µH, which is typical for an EV sized series wound motor, the amount the current rises during the 50µs delay can be calculated;

$$\begin{aligned} V &= L \frac{di}{dt} \\ di &= \frac{Vdt}{L} \\ &= \frac{50V \times 50\mu S}{500\mu H} \\ &= 5A \end{aligned}$$

Since the current sensing is for limiting purposes only and not to protect the power stage from short circuits, this delay is quite acceptable. The additional 5A of current should cause no problems with the power stage thermally. If the current needs to be limited to an exact value, the preset current limit could be lowered slightly to compensate for the overshoot.

To protect against a short circuit condition, the power stage would need to be switched off very fast, ideally within 10ns. It would be impossible for the microcontroller to respond in such a short time. Extra hardware could possibly be added to help prevent damage due to a short, such as crowbar circuits to direct current away from the power stage and blow the fuse.

The battery voltage and motor speed limits are less critical, and have a lower priority in the microcontroller's firmware. Consequently their response time is slightly longer. Figure 4.10 shows the response time to an over-speed condition. The test results for the voltage limits produced similar response times.



**Figure 4.10 – Speed Limit Response Time**

The captured waveform shows that it takes about 120 $\mu$ s for the microcontroller to switch off the PWM output in the event of an over-speed condition. The extra speed gained by the motor would depend on how much it is loaded; however even with no load it would be impossible for the motor to reach dangerous speeds in 120 $\mu$ s, so this response time is more than adequate. The delay also does not present a problem for voltage limiting, the voltage should not fall or rise to dangerous levels in a 120 $\mu$ s period if the drive was previously operating normally.

#### **4.6.8 Conclusions of Testing**

The testing proved that the microcontroller operated as desired and provided a control system that implements all the required features with adequate response times. Even though the accuracy and resolution of the ADC may not be particularly high, it is more than sufficient for the purposes of this design.

# Chapter 5

## **5.0 Power Electronic Design**

This chapter covers the specifics of the power electronic design based on the half bridge circuit discussed in Chapter 2. To ensure the design will function correctly and reliably, careful attention must be paid to the power electronic circuit design. When working at high powers, additional challenges are faced that otherwise do not exist in low power circuits. These include the issues of switching device selection, thermal design, EMI and EMC, and design layout. Actual construction and testing of a full power stage that will operate at high power levels would be a costly and time consuming exercise and is out of the bounds of this project. This chapter will cover the conceptual design of the major power electronic components, but they are not able to be fully constructed.

### **Section 5.1 – Selection of Power Switching Devices**

Electronic motor controls have only been made possible by the development of high power solid-state devices. Today, large transistors that have the ability to switch significant amounts of current at high speeds and voltages are readily available. The final design is intended to have a maximum motoring current limit of around 300A and a regeneration limit of 100A, so the part selection is based around these values.

#### **5.1.1 Power Transistors**

The most common power devices that have been used in motor drives are metal oxide silicon field effect transistors (MOSFET), insulated gate bipolar transistors (IGBT), bipolar junction transistors (BJT), and silicon controlled rectifiers (SCR, also known as thyristors). Early designs used BJTs and SCRs, but advancements in semiconductor technology have made these parts obsolete in this application. MOSFETs and IGBTs are now being used in modern drives. Table 5.1 shows the main attributes of each of the listed power devices.



Application	Power Device Technology	Attributes
High current Moderate voltage	MOSFET	<ul style="list-style-type: none"> <li>• Very low on resistance</li> <li>• Very high speed (500kHz)</li> <li>• Gate easily controlled</li> </ul>
High voltage High current	IGBT	<ul style="list-style-type: none"> <li>• Lower on losses than MOSFET</li> <li>• Average speed (20kHz)</li> <li>• Gate easily controlled</li> </ul>
High voltage Low current	BJT	<ul style="list-style-type: none"> <li>• Low on losses</li> <li>• High speed (100kHz)</li> <li>• Complex, expensive drive required</li> </ul>
High current Low voltage	SCR	<ul style="list-style-type: none"> <li>• Need transient suppression</li> <li>• Complex, expensive drive required</li> <li>• Hard to drive with PWM signal</li> </ul>

**Table 5.1 – Power Device Attributes**

SCRs are no longer normally chosen for motor drive systems as they cannot be easily turned off after they have been gated on. This makes it difficult to use a PWM signal to drive the motor. Power BJTs are also an uncommon choice for high power drives as this type of transistor is a current gain device. This means that to put the transistor into saturation (i.e. fully on), sufficient base current must be available. A base current requirement of one-tenth of the collector current is not uncommon (Valentine 1998), so this results in a design that dissipates a large amount of power in the gate drive circuit. However, BJTs are still commonly used for low power applications due to their low cost.

The power MOSFET offers high switching frequency and simple gate drive requirements. They can achieve fast switching speeds because they store no charge in their drain-to-source area. The MOSFET's gate is voltage controlled and requires a low average current. MOSFETs are available in a wide range of voltages (20V to 1000V) and currents (1 to 1000A). Typically, MOSFETs are used in motor drives rated less than 200V, otherwise the device has to become large to achieve a low on resistance (Valentine 1998). Nowadays, MOSFETs are quite inexpensive and are a popular choice for many switch-mode applications.

IGBTs offer performance similar to BJTs but are driven like a MOSFET. They are not able to switch as fast as MOSFETs due to charge stored in the device, however their speed is still adequate for most motor control applications. They are well suited to high voltage, high current applications and are common in 3 phase line powered motor drives. Since this design operates from a 48V nominal battery pack, N-channel MOSFETs are the most suitable switching device due to their low losses, easy to drive gate and high current rating.

### 5.1.2 MOSFET Selection

With the decision made to use N-channel MOSFETs as the power switching devices, a suitable part, or combination of parts, must be decided upon. In the interests of economy, it is more economical to use a bank of smaller transistors to handle the required current rather than using a single large device. Power MOSFETs are suitable for parallel operation because of their negative temperature coefficient, which helps them to share the load current evenly between the devices. When one device starts to heat up, its resistance increases (i.e. a negative temperature coefficient), so some of its current is taken by another device with a lower resistance. This effect allows the devices to current share when connected in parallel, with a low risk of thermal runaway. To help ensure that each device's characteristics are similar enough to safely current share and operate at the same temperature, it is preferable that all the transistors come from the same production batch as they are more likely to be closely matched.

There is a huge range of power MOSFETs available, and there is more than one particular part that would be suitable. There are many parts from different manufacturers that have similar specifications. Choosing the most cost effective combination of parts can be difficult due to the wide range available. When choosing a part, one must consider if it is better to use a large number of low current parts or a few high current parts. Generally, it is cheaper to use more low current parts, but this can cause trouble with current sharing and gate drive when large numbers of transistors are involved.

The voltage rating of the transistor must be high enough to ensure that it is not damaged by transients or spikes. Another constraint is the power handling capability of the transistor. The device will have a specified maximum power dissipation at a certain temperature. This must not be exceeded or the device could be damaged. In addition to the ratings and specifications of the transistors, their physical packaging and how they can be mounted, insulated, and cooled also need to be considered.

In light of these issues and the required design specifications, the IRF1407 MOSFET from International Rectifier was chosen. These are an avalanche rated TO-220 packaged device rated at 75V, 130A with an on resistance of  $0.0078\Omega$ . It should be noted, however, that the TO-220 package has a limitation of 75A maximum. Since the drive has a higher current limit in the motoring (buck) mode, the high side of the bridge has 7 of these MOSFETs and the low side has 3, making a total of 10. Section 5.2 will deal with the loss calculations to ensure the devices will operate reliably.

### 5.1.3 MOSFET Power Device Attributes

The MOSFET's data sheet specifies many parameters and characteristics of the device. It is important that the design operates the MOSFETs within their limitations, as described by the data sheet. While MOSFETs are fairly robust power devices, to design a reliable circuit the parts must be operated well within the limits on the data sheet, which are usually a worst-case maximum rating. Using the absolute maximum design values as normal design values will lead to unreliable, short lived operation. It is suggested that the design operates within 40-60% of the device's maximum (Valentine 1998) to help ensure reliability of the power stage. The specifications that are most important to a motor control application are listed below.

Maximum drain-to-source voltage – the internal body diode determines the breakdown voltage of the transistor, and this rating has a positive temperature coefficient. This is critical in motor drive applications as the inductive motor load and fast switching times cause voltage spikes. MOSFETs rated for avalanche operation are able to momentarily exceed the breakdown voltage. If a spike sends the diode into breakdown, current flows into the device and the energy must be absorbed.

Maximum continuous drain current – this is the current rating that will raise the device's junction temperature to its rated maximum while its reference temperature is held at 25°C. This can be calculated by the following equation:

$$I_D = \sqrt{\frac{P_D}{R_{DS(on)} \text{ at } T_{j,max}}}$$

This rating is not particularly useful in a practical design, as it is almost impossible to keep the case at 25°C while operating at high power due the losses. The data sheet has graphs that illustrate how the drain current is de-rated at high power dissipations.

Maximum pulsed drain current – this is the maximum allowable peak drain current that the device can safely handle for a 10µs pulse. This rating takes into account the device's physical limitations such as wire bond and metal properties. The duty cycle for the rating is very short, typically 10µs, 1s off. If the device is to survive a short circuit motor condition then some time of current limiting that responds in less than 10µs would be necessary.

Typical and maximum drain-to-source on resistance – this is the DC resistance between the drain and source terminals with a specified gate-to-source voltage applied to bias the device into the on state. This parameter has a positive temperature coefficient. The maximum  $R_{DS(on)}$  value should be used when calculating power losses. This resistance is directly affected by junction temperature and gate-to-source voltage.

Gate threshold voltage – This is an important parameter when designing the gate drive circuit. It is the gate-to-source voltage ( $V_{GS(th)}$ ) at which current begins to flow from the source to the drain. The threshold voltage has a negative temperature coefficient, which can lead to problems at high temperatures.

Gate charge – the specifications pertaining to gate charge are used to design the gate drive circuit and estimate switching speeds and switching losses.  $Q_G$  is the total gate charge required to charge the device's input capacitance to the applied gate voltage.  $Q_1$  is the charge required to charge the input capacitance to the  $V_{GS(on)}$  required to maintain the test current  $I_D$ . The time taken to charge the capacitance is known as the turn on delay time.  $Q_2$  is the charge time required for the drain-to-source voltage to drop to  $V_{DS(on)}$

Maximum drain-to-source avalanche energy –  $E_{as}$  defines the maximum allowable energy that the device can handle in the event of an inductive current spike. It is tested at the rated  $I_D$  of the transistor and is a single pulse, non-repetitive condition. This rating has a negative temperature coefficient. For long term reliability, the motor control should not operate the MOSFET in avalanche mode frequently, only during a random voltage spike event.

Maximum power dissipation –  $P_D$  specifies the maximum power the transistor can dissipate which takes the junction temperature to its maximum rating while the reference temperature (i.e. the case) is held at 25°C. It is calculated by:

$$P_D = \frac{T_{j,max} - T_r}{R_{thjr}} \text{ where } R_{thjr} \text{ is the thermal resistance of the junction}$$

Once again, keeping the case at 25°C is very difficult, so this rating must be used with caution.

Maximum junction temperature – The  $T_{j,max}$  value represents the maximum allowable operating temperature of the silicon junction in the transistor. Exceeding this temperature will reduce the life of the transistor. Some studies suggest that when operating above 100°C, the device's life is doubled for every 10°C reduction in temperature.

#### 5.1.4 Other Devices

In addition to the transistors, diodes are also required to carry the motor current when the PWM signal is in the off state. MOSFETs have a built in parasitic diode that is rated at the full drain current of the transistor. This body diode will be used to carry the freewheel current in the low-side of the half bridge and the regeneration current in the high-side of the half bridge. Since the body diode in the MOSFET does not have a low forward voltage drop, adding external diodes with a lower voltage drop will improve performance, however it is expected that the body diode will be sufficient for the purposes of this design.

The specifications for the MOSFET's internal body diode are given in the data sheets. Two important parameters that affect the use of the diode are:

Diode forward on voltage – the diode's voltage drop when forward biased is particularly important if it is used as a freewheeling diode in a motor control circuit. It will determine the loss associated with this mode of conduction. A higher voltage drop will cause more loss.

Diode reverse recovery time – the reverse recovery time of the diode is important, a fast recovery will produce sudden changes in current, which increase EMI. The body diode of a MOSFET tends to have a fast recovery time, which may not be desirable in all applications.

## Section 5.2 – Thermal Design

This section deals with the calculations of power dissipation and losses to ensure the devices operate within specification. Losses in both the high and low side of the bridge are calculated, as well as the diodes. Cooling requirements are also addressed, discussing the use of heatsinks and forced cooling. Thermal design is a complex topic and a complete analysis of all issues involved is outside the scope of this project. Topics such as thermal cycling, failure mode analysis and thermal modelling could not possibly be fully covered here. However, the basics will be covered and this should be sufficient for the particular design presented here.

### 5.2.1 Thermal Issues

Apart from gate damage, heat is ultimately the cause of all transistor failures; there are three main events that can invoke a power device to fail due to heating;

- An electrical overstress of the device
- High peak power cycling caused by low PWM duty cycles
- Extreme temperature cycling caused by a repetition of large temperature swings

Electrical overstress failures are caused by operating the device outside of its safe operating area (SOA), usually due to current surges, voltage spikes or other unexpected events. Sometimes the failure can be spectacular, totally obliterating the device. The device fails because the heat generated by the event cannot be removed quickly enough to prevent a damaging rise in temperature. This is one reason current limiting circuits are very important.

Power cycling failures and thermal cycling failures are somewhat similar. A power cycling failure involves mechanical and thermal stresses on the device experienced due to a low PWM duty cycle. In this situation, the external case temperature of the device may not be elevated, but the failure is caused by localised heat on the semiconductor die. Localised heating in the device junction can crack the device or lift the wire bonds that carry the current from the silicon die due to heating above the melting point of solder.

Thermal cycling failures are usually caused by large temperature variations of the device itself. These may be caused by heat produced by the device in operation or environmental conditions. Changes in temperature cause materials in the device to expand and contract. Due to the range of materials used in the device, some parts have different coefficients of thermal expansion to others, so some expand and contract faster or slower than others. This cyclic expansion and contraction can cause stresses at the point of interface between two different materials. The faster the change in temperature the greater the stresses caused. Eventually the device will fail due to degradation of its ability to meet its original specifications.

To maximise reliability and help prevent these failure modes, it is important that careful attention is paid to the thermal management of the power devices. The devices need to be kept as cool as possible and sudden temperature changes need to be avoided. In normal operation, the amount of power lost as heat can be estimated using specifications from the data sheets, and this will determine the amount of cooling required.

### **5.2.2 Thermal Requirements and Models**

After the devices for the power stage and the control method have been defined, the thermal requirements of the design based on the final application can be considered. Along with the expected power dissipation under normal operating conditions, the expected ambient temperatures and possible abnormal operating conditions must also be considered. In determining the size of the heatsink required, worst-case conditions should be assumed. The primary factors that define the heatsinking requirements are;

- Conduction losses in the power devices
- Switching losses in the power devices
- Maximum junction temperature and device power dissipation

Cost is also an issue when designing a thermal management system. The cost of the power electronics can be minimised with the use of a highly effective heat removal system, however this will increase the cost of the thermal system. The availability of materials and ease of construction will also affect the design. Thus, a balance is required depending on the requirements of the design and resources available.

To model the thermal characteristics of the design, the system can be analysed during its steady-state condition. In this state, it must be ensured that the heat generated by the power device is equal to the heat carried away. An analogy can be made between the conduction of electricity and the conduction of heat, just as electrical resistance is associated with the conduction of electricity, thermal resistance is associated with the conduction of heat. Figure 5.1 illustrates this heat conduction model. The basic model shows conduction through a slab of material, the temperature distribution through the slab is linear and can be modelled by a resistance. To decrease thermal resistance, it is desirable to minimise the length of the thermal path ( $L$ ) and maximise its area ( $A$ ). The illustrated thermal model of a power stage shows how the various thermal resistances add together to make a total thermal resistance. This model can be used to calculate the junction temperature of the transistor given the thermal resistance of the various parts and the ambient temperature.

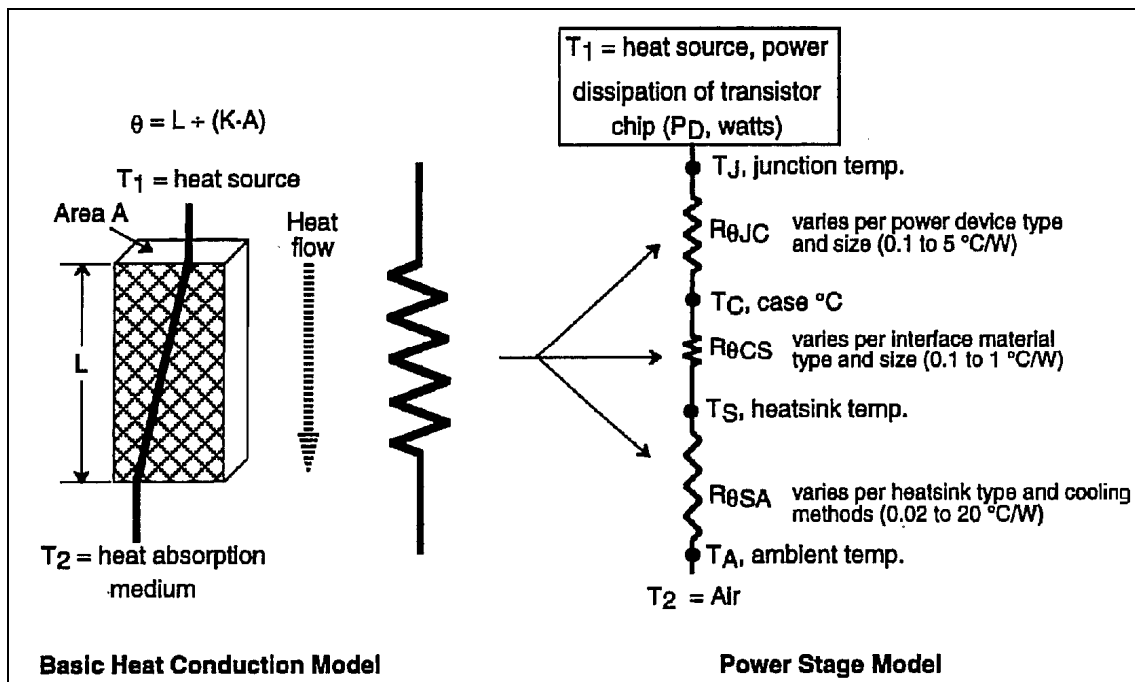


Figure 5.1 – Basic Heat Conduction Model (Valentine 1998)

Data sheets give specifications of the thermal resistance of the semiconductors junction-to-package ( $R_{\theta JC}$ ), which gives a measurement of the devices ability to conduct heat away from the junction. In addition to this there is a thermal resistance of the material that interfaces the transistor to the heatsink and another for the heatsink to the ambient air (or other cooling system).



### 5.2.3 Loss Calculations

The main source of loss in the power transistors can be attributed to the drain-to-source on resistance of the MOSFETs ( $R_{DS(on)}$ ), this is known as conduction loss. In addition to this loss, there is also a switching (or dynamic) loss caused by the finite time it takes the transistor to turn on and off. The conduction losses can be calculated by the following formula:

$$P_D = I_D^2 R_{DS(on)} D, \text{ where } D \text{ is the duty cycle of the transistor}$$

The worst case dissipation from the MOSFETs occurs at 100% duty cycle or when the drive is in current limit. A current limit of 300A in the motoring mode will be assumed for this design, which is the desired maximum current in the final design for the electric vehicle. The current limit in regeneration mode will be limited to a lower value. Maximum conduction losses for the high side bank of MOSFETs at this current can thus be calculated;

$$\begin{aligned} P_D &= I^2 \times R_{DS(on)} \\ &= 300^2 \times 0.014 \\ &= 702W \end{aligned}$$

There are 7 MOSFETs in the high side of the bridge, this equates to 100W per device. This power is within their maximum power dissipation rating of 330W. At 300A, the MOSFETs will be carrying around 43A each, which is well below their theoretical maximum of 130A

The regeneration current will be limited to 100A to protect the batteries and the motor. The conduction losses in the low side of the half bridge can be calculated the same as the high side;

$$\begin{aligned} P_D &= I^2 \times R_{DS(on)} \\ &= 100^2 \times 0.0078 \\ &= 78W \end{aligned}$$

There are 3 MOSFETs in the low side of the bridge, this equates to 26W per device. This power is well within their maximum power dissipation rating of 330W.

This calculation is sufficient for determining the suitability of the power devices, however, the actual value of  $R_{DS(on)}$  depends on the temperature of the device. As mentioned previously,  $R_{DS(on)}$  increases with temperature, so to calculate the losses at operating temperature, it is necessary to include the temperature coefficient of  $R_{DS(on)}$  in the equation for conduction loss.

Hence, the equation for conduction losses at a specific temperature is:

$$P_D = I_D^2 R_{DS(on)} (1 + \delta \Delta_T) D$$

where  $\delta$  is the temperature coefficient of  $R_{DS(on)}$  in  $^{\circ}\text{C}^{-1}$

However, the dynamic losses need to be considered in addition to the conduction losses. Dynamic losses are more complex to calculate, as they depend on the characteristics of the MOSFET and the gate drive circuit. Dynamic losses are caused as the MOSFET passes through the linear region as it is switched on and off by the PWM signal.

The time the MOSFETs spend in the linear region depends on the gate capacitance and the gate drive's output impedance. These define the rise and fall times of the MOSFET's switching. The rise and fall times ( $t_r$  and  $t_f$ ) can be estimated by the following calculations:

$$t_r = t_f = \frac{Q_G}{I_G} + \frac{L_{LUMP} I_G}{V_{GS} - V_{TH}}$$

where  $V_{GS}$  is the gate-to-source voltage,  $V_{TH}$  is the gate turn on threshold,  $L_{LUMP}$  is the parasitic inductance of driver to MOSFET connection,  $Q_G$  is the total gate charge, and  $I_G$  is the gate drive current (Mappus 2003). Once the rise and fall times are known, the dynamic loss can be calculated by:

$$P_D = \frac{1}{2} I_D V_S (t_r + t_f) f$$

where  $f$  is the frequency of the PWM signal and  $V_S$  is the input source voltage and  $I_D$  is the drain current (Mappus 2003).

In the case of this design the dynamic losses can be calculated using values from the MOSFET and gate drive IC data sheets. A parasitic inductance of 100nH will be assumed, so the high side dynamic losses can be calculated as follows;

$$t_r = t_f = \frac{7 \times 160 \text{ nC}}{3 \text{ A}} + \frac{100 \text{ nH} \times 3 \text{ A}}{12 \text{ V} - 4 \text{ V}}$$

$$= 41 \text{ nS}$$

$$P_D = \frac{1}{2} 300 \times 48 (41 \text{ nS} + 41 \text{ nS}) 15 \text{ kHz}$$

$$= 88.7 \text{ W}$$

This is around 12.7W per device due to switching loss, adding this to the conduction loss gives around 113W per device. This is still within specification and allows some headroom for unexpected events.

Once again, the low side dynamic losses can be calculated in the same way. They will have lower rise and fall times because the gate drives do not need to charge as many gates (4 as opposed to 10), this will reduce the dynamic losses.

$$\begin{aligned}
 t_r = t_f &= \frac{3 \times 160 \text{ nC}}{3 \text{ A}} + \frac{100 \text{ nH} \times 3 \text{ A}}{12 \text{ V} - 3.5 \text{ V}} \\
 &= 197.5 \text{ nS} \\
 P_D &= \frac{1}{2} 100 \times 48 (197.5 \text{ nS} + 197.5 \text{ nS}) 15 \text{ kHz} \\
 &= 14 \text{ W}
 \end{aligned}$$

This is around 5W per device due to switching loss, adding this to the conduction loss gives around 31W per device. It may seem that since the losses are so low, less transistors would be required in the low side. However, since the parasitic diodes in the low side transistors are used as freewheeling diodes for the high side, their losses will in fact be higher.

In addition to losses in the power transistors, there are also losses in the freewheeling diodes. The worst case scenario for the freewheel diodes in the low-side of the bridge is when the circuit is operating at 50% duty cycle, this is an average current of 150A. The loss can be estimated by the forward voltage drop of the diode and the current flowing through it.

$$P_D = V_{SD} \times I_D$$

where  $V_{SD}$  is the diode forward voltage and  $I_D$  is the diode current..

In the case of this design, the maximum diode loss is approximately;

$$\begin{aligned}
 P_D &= 1.3 \text{ V} \times 150 \text{ A} \\
 &= 195 \text{ W}
 \end{aligned}$$

There are 3 diodes in the low side of the bridge, so this equates to 65W per device.

The worst case loss for the high side diodes occurs when they are carrying the full regeneration current to the batteries, i.e. 100A. The diode loss is calculated by;

$$\begin{aligned}
 P_D &= 1.3 \text{ V} \times 100 \text{ A} \\
 &= 130 \text{ W}
 \end{aligned}$$

There are 7 diodes in the low side of the bridge, so this equates to 13W per device.

Therefore, the total calculated power dissipation in the switching devices in the motoring mode when operating at a current limit of 300A is 113W per transistor, and 65W per diode. This is a total of 791W transistor loss and 195W diode loss. In the regenerating mode, the losses are 31W per transistor and 19W per diode. This is total of 92W transistor loss and 130W diode loss. At these power levels, the power devices will definitely require mounting on a heatsink of some sort. The heatsinking and cooling systems must be able to dissipate this amount of heat without allowing the junction temperature to become too high and causing damage to the device.

While the losses may at first seem rather large, when they are compared to the output power they are in fact quite small. For example, when the drive is operating at its peak power output of 300A, this gives an output power of around 14,440W. The efficiency at this operating point is about 95%, which is very good. Similarly, when operating at 50% duty cycle the efficiency is 92%.

#### 5.2.4 Cooling Requirements

There are three modes that allow heat transfer in a system: conduction, convection, and radiation. Conduction occurs by diffusion through solid material or stationary liquid or gas. Convection involves the movement of a liquid or gas between two points. Radiation occurs through electromagnetic waves. The interior of a semiconductor device uses conduction, while a heatsink will use both radiation and convection to dissipate heat.

As discussed previously, the main components of a power stage's thermal system are the semiconductor device, an interface material and the heatsink itself. The thermal resistance between the semiconductor's case and the heatsink is governed by the interface material. This can vary according to the mounting pressure of the device, the type of material used, and changes in the material over time. The thermal resistance of the heatsink will change depending on the air (or perhaps liquid) flow velocity and turbulence.

Once the thermal resistance of the power stage model and the maximum power dissipation are known, an approximation of the required heatsink thermal ratings can be made to ensure the junction temperature of the devices do not exceed their ratings. The junction temperature can be calculated by simply adding the ambient temperature to the sum of the thermal resistances multiplied by the power dissipation;

$$T_j = PD(R_{\theta_{jc}} + R_{\theta_{cs}} + R_{\theta_{sa}}) + T_a$$

where  $T_a$  is the ambient temperature,  $R_{\theta_{jc}}$  is the junction-to-case resistance,  $R_{\theta_{cs}}$  is the case-to-heatsink resistance, and  $R_{\theta_{sa}}$  is the heatsink-to-air resistance.

A value of  $R_{\theta_{jc}}$  is provided by the semiconductor manufacturer, and depends on the package type used and the internal structure of the device. In the case of the IRF1407, it is  $0.45^\circ\text{C}/\text{W}$ .

$R_{\theta_{cs}}$  and  $R_{\theta_{sa}}$  need to be determined from the heatsink and cooling method used.  $R_{\theta_{cs}}$  is calculated according to the interface material chosen, thickness of material, and area of the power device;

$$R_{\theta_{cs}} = \rho \frac{t}{A}$$

where  $\rho$  is the thermal resistance of the interface material,  $t$  is the average material thickness, and  $A$  is the contact area of the power device.

There are types of interface material, some of which electrically insulate the semiconductor's case from the heatsink while others allow electricity to conduct. A list of typical materials and their thermal resistance  $\rho$  are given in table 5.2. Silicone grease allows the best conduction of heat, but does not electrically insulate the transistor from the heatsink, so it becomes live. This may or may not cause a problem, depending on the design, physical layout and housing of the circuit. The contact area of a TO-220 package is around  $1.5\text{cm}^2$ , using silicone grease gives a  $R_{\theta_{cs}}$  of around  $0.5^\circ\text{C}/\text{W}$

Material Type	Resistance $\rho$ ( $^\circ\text{C}\cdot\text{cm}/\text{W}$ )	Thickness (cm)	Conductive
Silicone grease	15.24	0.0012-0.0051	Yes
Mica	16.76	0.00508	No
Silicone rubber	20.57	0.127-0.635	No
Mylar	59.94	0.0013-0.0051	No
Dead air	304.8	-	No

**Table 5.2 – Interface Material Properties**

There is a wide range of heatsinks available, which come in various sizes, designs and cooling methods. Table 5.3 lists the most popular heatsink types and their attributes.

Heatsink Type	$R_{\theta sa}$ Lower Limit (°C/W)	Cost	Notes
Liquid cooled	0.007	Highest	Pump, radiator and fan needed
Bonded fin (with forced air)	0.028	High	Fan needed
Custom forced air extrusion	0.092	Medium	Fan needed, special order
Bonded fin (no forced air)	0.22	Medium	Has thin fins
Extrusion aluminium heatsink	0.6	Low	Catalogue stock
Flat extrusion	1.4	Lowest	Cut to length
Standard aluminium heatsink	4.2	Low	Catalogue stock

**Table 5.3 – Heatsink Thermal Resistances (Valentine 1998)**

The total thermal resistance of the system can be calculated using the maximum desired junction temperature, maximum ambient temperature and the total power dissipation;

$$R_{\theta TOTAL} = \frac{T_{max} - T_a}{P_D}$$

From this total thermal resistance, the required thermal resistance of the heatsink can be calculated by subtracting the total away from the junction and interface thermal resistances;

$$R_{\theta sa} = R_{\theta TOTAL} - R_{\theta jc} - R_{\theta cs}$$

Since multiple MOSFETs are used in parallel, their combined thermal resistance can be calculated in the same way as the equivalent resistance of two or more paralleled resistors.

A fan can be used to further reduce the temperature of the heatsink and can make a significant improvement its performance; the addition of fans can easily halve the effective thermal resistance of a heatsink. If the required fan is not specified in the heatsink data, it can be estimated without using complex fluid dynamics. If the heat to be dissipated is known the fan size can be estimated because the overall system air temperature differential above the inlet ambient is directly proportional to the heat dissipated.

It is possible to estimate the amount of required cooling air, which will determine the required airflow of the fan. The basic heat transfer equation is:

$$Q = mC_p \Delta T$$

where Q is the amount of heat transferred to system (watts),  $C_p$  is the specific heat of air (J/kg), m is the mass flow rate of air (kg/s) and  $\Delta T$  is the air temperature differential (cabinet to ambient outside air).

The relationship between mass flow rate and volumetric flow rate is;

$$m = \rho G$$

where  $\rho$  is air density (kg/m<sup>3</sup>) and G is the volumetric flow rate (m<sup>3</sup>/s).

The required volumetric flow rate is then calculated from;

$$G = \frac{Q}{\rho C_p \Delta T}$$

This yields a rough estimate of the airflow needed to dissipate a given amount of heat at sea level. It should be noted that it is the mass flow rate of air, not its volumetric flow rate, which governs the amount of cooling. In addition, the actual flow required will be greater than the calculated number due to wind resistance in the enclosure.

A heatsink (or combination of heatsinks) must be chosen for this design, and it needs to meet the following criteria;

- Be able to dissipate sufficient heat to prevent dangerous junction temperatures under all expected operating conditions.
- Be of suitable physical size and construction to allow mounting of the power devices and fit into a reasonably sized enclosure.
- All devices in each side of the half bridge must be bolted onto the same heatsink to promote even current sharing between the devices.
- Electrical isolation is required between the high and low side power devices, as well as the chassis of the design.

In light of these criteria, it was decided that the high and low side of the bridge would be mounted on separate heatsinks. A large heatsink for the high side, and a smaller heatsink for the low side. To reduce the thermal resistance of the power device to the heatsink, silicone grease will be used as the interface material.

This has the lowest thermal resistance but has the disadvantage of not electrically insulating the transistors from the heatsink. This means that the high and low side heatsinks must not touch each other or the chassis to avoid a short circuit. This could be achieved by mounting them on a plastic or wooden board, or using plastic mounting standoffs in a metal case.

Another factor to consider is that the drive will not be continuously operating at its 300A limit. This is a peak current that should only be drawn if the vehicle is accelerating hard or climbing a steep hill. This means that it is possible to use a smaller heatsink than otherwise would be required to dissipate full power continuously.

To calculate the required thermal resistance of the high side heatsink for the maximum estimated power loss of 791W, a maximum ambient temperature of 30°C and maximum junction temperature of 165°C is used (the device's maximum allowable junction temperature is 175°C);

$$\begin{aligned} R_{\theta TOTAL} &= \frac{165 - 30}{791} \\ &= 0.171^\circ\text{C}/\text{W} \end{aligned}$$

Since multiple MOSFETs are mounted on a single heatsink, the thermal resistance of the junction-to-case and case-to-heatsink of each device must be added in parallel. In this case,  $R_{\theta jc}$  is 0.45°C/W and  $R_{\theta cs}$  is 0.5°C/W, and there are 7 MOSFETs in parallel. The required thermal resistance of the heatsink can then be calculated;

$$\begin{aligned} R_{\theta sa} &= R_{\theta TOTAL} - \left( (R_{\theta jc} + R_{\theta cs})^{-1} \times n \right)^{-1} \\ &= 0.171 - \left( (0.45 + 0.5)^{-1} \times 7 \right)^{-1} \\ &= 0.035^\circ\text{C}/\text{W} \end{aligned}$$

This is a very low thermal resistance, so a high quality heatsink must be used. Farnell have a range of heatsinks available with low thermal resistances. The type chosen for this design is a compact tunnel style bonded fin heatsink with forced cooling and is manufactured by IMI Marston, Farnell catalogue number 311-0837. Its dimensions are 120x250x125mm (HxWxL). The heatsink manufacturer specifies that the heatsink will have a thermal resistance of 0.05°C/W when forced cooled with two 120mm fans mounted directly on the heatsink with approximately 3m<sup>3</sup>/min of flow each. This is higher than the calculated 0.035°C/W calculated above, but it is still expected to be sufficient when considering the average power dissipated will be much less than the maximum. Experimental testing would need to be carried out to determine if this is in fact sufficient.



For the low side, a smaller heatsink will be required since the losses are much lower. The maximum loss in the low side occurs at 50% duty cycle in the motoring mode when the diodes are dissipating 195W. The calculations are as follows;

$$R_{\theta_{TOTAL}} = \frac{165 - 30}{195}$$

$$= 0.692^{\circ}C/W$$

Once again the MOSFETs are mounted on a single heatsink, so the required heatsink thermal resistance can be calculated;

$$R_{\theta_{sa}} = R_{\theta_{TOTAL}} - \left( (R_{\theta_{jc}} + R_{\theta_{cs}})^{-1} \times n \right)^{-1}$$

$$= 0.692 - \left( (0.45 + 0.5)^{-1} \times 3 \right)^{-1}$$

$$= 0.38^{\circ}C/W$$

The heatsink chosen for the low side is available from Jaycar Electronics (HH-8530) and is of similar design to the one chosen for the high side, however it is slightly smaller and is a simple aluminium extrusion. Its dimensions are 84x250x84mm (HxWxL).

It is also of the tunnel style, and with a single 80mm fan with a flow rate of 1m<sup>3</sup>/min will have a thermal resistance of approximately 0.35°C/W. This is sufficient to keep the low side cool under worst case conditions. This heatsink cannot be underrated because prolonged operation at around 50% duty cycle is highly possible, thus a continuous dissipation of 195W must be accounted for.

It should be noted that while the calculations presented give a starting point for the thermal design of the power stage, actual physical experimentation is required to ensure that the drive operates within the specifications of the semiconductor devices. This would include measuring the temperature rise of various parts while they are under load and determining the continuous and peak operating power levels.

### 5.2.5 Other Issues

There are other topics in thermal design that are simply out of the scope of this project. However, some features that could fairly easily be implemented in this design include thermal sensing and synchronous rectification. Since the design utilises a microcontroller, adding a thermocouple or other temperature sensor to the power stage (and perhaps the motor) could allow the circuit to sense an over temperature condition and reduce the output power or shut down.

One way to further reduce losses is using a technique called synchronous rectification. The concept uses a MOSFET to act as the freewheeling diode. The forward voltage drop of a switched on MOSFET is much lower than that of even a Schottky rectifier. The resulting improvement is significant, especially in low output voltage converters. In order to achieve synchronous rectification during MOSFET switching, an alternate triggering control is needed to turn the two switching transistors on and off alternately. In simple terms, synchronous rectification is achieved by turning on the low side of the bridge when the high side is switched off (or vice-versa for regeneration). The low on-resistance of the synchronous MOSFET reduces the losses that would normally occur with a freewheeling diode. Careful attention must be paid to the design to ensure that a shoot through condition does not occur.

## Section 5.3 – Layout, EMI and EMC Issues

In a motor drive, layout is a critical part of the design. The layout of parts can mean the difference between a circuit that works properly and one that is unreliable or even does not work at all. Due to the high rates of  $di/dt$  and  $dv/dt$  involved in a PWM motor control, large amounts of noise, as well as current and voltage spikes are produced. These can have damaging effects on components or produce undesired operation of the circuit if they are not properly accounted for in the design. In a worst case scenario, the entire power stage could be destroyed due to noise in the gate drive circuit.

### 5.3.1 Power Stage Layout

The main objectives pertaining to power stage layout are;

- Sufficiently control noise at the gate drives so that the power stage is not turned on or off erratically.
- Minimise the radiated noise with good layout practices

Gate drive noise management is achieved mainly through good layout of the gate drive circuit. Gate drives are susceptible to parasitic coupling, particularly in the high side where the gate driver has a high  $dv/dt$  with respect to power ground due to the bootstrapping of the gate drive voltage. If the source terminal of a power device is used as a ground plane for its respective gate drive, then parasitic coupling to power ground is minimised which increases the noise immunity of the gate drive signal.

Another method for improving the noise immunity of the gate drive is minimising the loop area of the circuits. This involves routing circuit board traces close to or on top of their return, for example routing the gate drive signal on the top layer of the board and the return on the bottom layer directly beneath it. This reduces capacitive coupling and antenna effect that inject noise into the gate driver. Other ways to reduce noise in this area of the circuit include using optocouplers and a separate gate drive IC for each transistor.

Other areas in the power stage benefit from good layout, minimising loop areas is the main method of reducing the amount of noise radiated by the power stage. Power and return traces should be run on top of each other wherever possible. Right angle PCB traces should also be avoided as these can produce reflections and act as antennas, instead curved or 45° angles should be used.

### 5.3.2 Protection Against Noise and Spikes

Snubber circuits should be fitted at various points in the power stage to reduce radiated noise and protect components from voltage and current spikes. To minimise spikes due to the inductance of battery cables, a capacitive filter should be fitted across the high current power supply as close to the power stage transistors as possible. The filter should consist of a low capacitance (around  $2\mu\text{F}$ ) high frequency metallized film capacitor and a large (several thousand  $\mu\text{F}$ ) low ESR (equivalent series resistance) electrolytic capacitor. The connections between these capacitors and the power stage should be kept as short as possible to minimise stray inductance. Figure 5.2 illustrates a good power stage layout in contrast to a poor layout, note that the desirable layout is also the most compact.

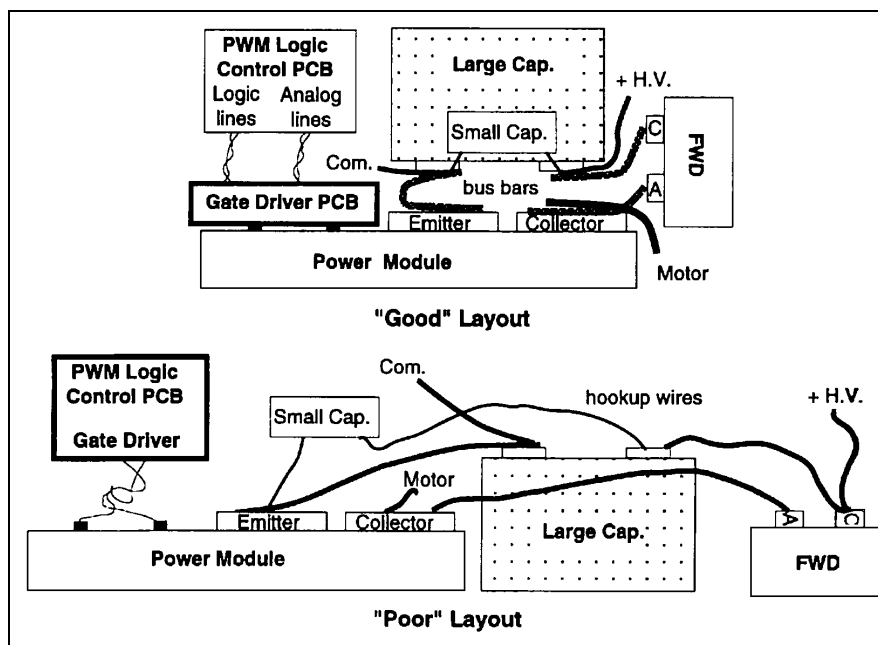


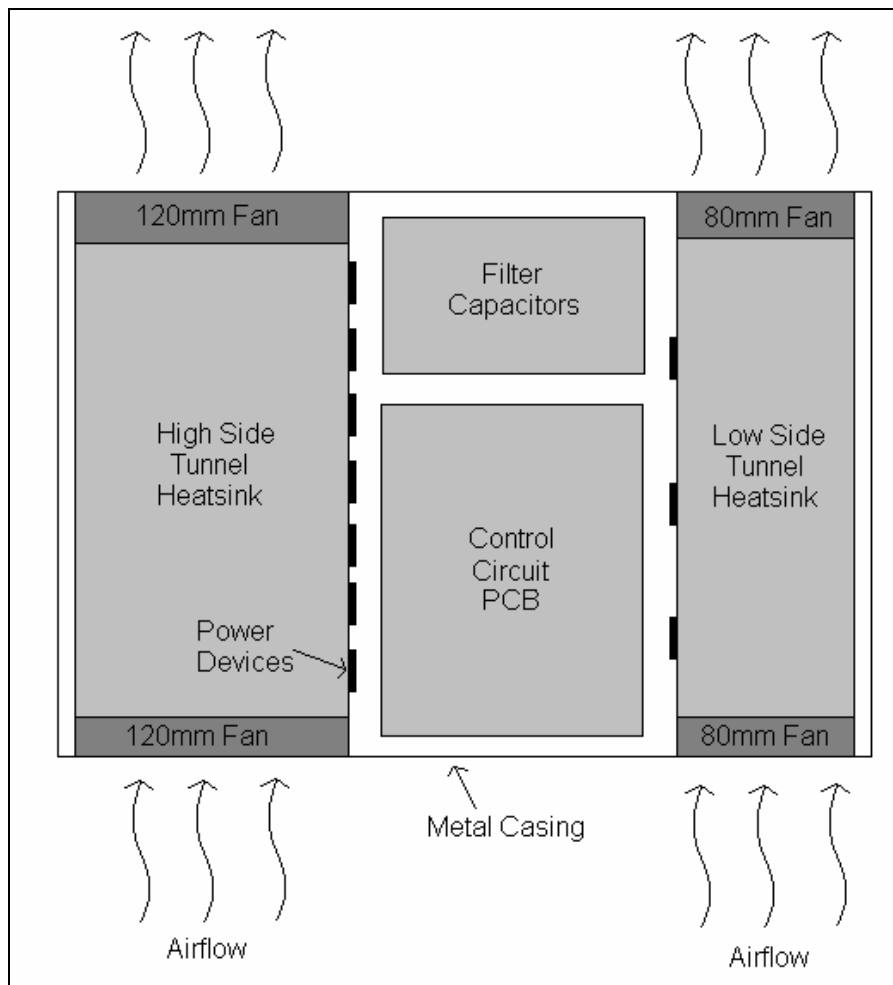
Figure 5.2 – Motor Controller Layout (Valentine 1998)

The MOSFETs and diodes should be protected against voltage spikes by voltage clamping networks such as MOVs (Metal Oxide Varistor) or transorbs. Snubber networks can also be fitted to minimise noise spikes due to the inductive kickback of the motor when it is switched off. The rise and fall time of the gate drive signal can be slowed to reduce high frequency harmonics from the switching signal. It should be noted, however, that slower rise and fall times reduce EMI, but increase switching losses.

Even after careful design, extensive testing of the power stage will be required to ensure desired operation. After testing, unexpected problems will no doubt arise, which will require further refinement to the design.

### 5.3.3 Power Stage Assembly

The entire circuit, including the two tunnel heatsinks, is to be housed in a metal enclosure as pictured in Figure 5.3. The heatsinks are electrically isolated from the metal casing by nylon mountings. The fans are mounted on each end of the heatsinks and force air through them. This air flows through the heatsinks but does not escape into the enclosure as the fans are sealed via gaskets. This way, the electronics will be protected from dust and humidity which otherwise might be drawn in by the fans.



**Figure 5.3 – Enclosure Layout**

A circuit diagram of the power stage is shown in Appendix D. The diagram shows the parallel connection of the MOSFETs and their associated circuitry. The circuit includes 15V zener diodes on the MOSFET gates to prevent accidental switch-on or damage to the gates in the event of a voltage spike. Transient protection is also fitted to the MOSFETs by means of MOVs, and capacitive filters are used to combat parasitic inductance from cabling and connections. The power stage is fused to protect against fire in the event of a short circuit condition.

# Chapter 6

## 6.0 Conclusion

This chapter gives final conclusions with respect to the overall project, discussing the achievement of objectives and the potential for further work in a number of areas. Undertaking this project, although difficult at times, has been a very rewarding and educational process. Many of the topics covered and skills learnt are useful over a wide range of electronic applications and are not limited to power electronics alone.

### **Section 6.1 – Achievement of Objectives**

The aims and objectives set out in the Project Specification at the beginning of the project were:

1. Select a suitable power electronic circuit for 4-quadrant operation.
2. Carry out a conceptual design for the control of the drive including speed control and protection.
3. Select and design the hardware for design implementation.
4. Write appropriate software for the drive.
5. Construct and test the drive.

Achieving these objectives was found to be much more time consuming than expected, nor was there sufficient time and resources available to fully construct the entire drive as a complete system ready for implementation into a vehicle. To completely test the system a PCB design would be required, along with a full sized power stage, both of which must address all the layout and EMI issues discussed in previous chapters. This would involve many stages of testing and revision, along with the high financial cost such testing incurs.

Nevertheless, objectives 1 to 4 were successfully carried out in full. Step 5 (construct and test) was completed but a full-size power stage was not built. However, the entire control system was constructed and tested, along with its other associated elements that form the design. These include gate drive circuits, current sense circuits, tachometer circuits, RS232 communications, power supplies, protection logic, along with the writing of associated software for the microcontroller and PC link software. The control system successfully implements the entire feature set discussed in Chapter 1.

Testing of the control system and its associated circuitry proved that the design is suitable and able to provide safe operation of the power stage. The features provided by the adjustable speed drive are well suited to application in an EV, particularly the fully adjustable parameters, which allow for safe operation of the EV during forward and reverse motoring as well as braking.

The completion of this project has demonstrated the advantages of using a microcontroller as the basis for the design and proved the feasibility of using a highly integrated approach to the circuit design. It has also highlighted the benefits of using software to incorporate many useful features in the design with a low parts count and minimal cost.

## Section 6.2 – Further Work

This project presents many opportunities for further work. The most obvious being the full construction and testing of a power stage to be coupled to the control system. This would be a large task, and involves a large amount of testing and careful design. The end result, however, would be a very useful design which could be used in many different types of electric vehicles. To reduce pollution and our demand on ever depleting fossil fuels, more environmentally sound ways of transport need to be developed. An EV motor control can be powered with high technology batteries or hydrogen fuel cells to make a long range road-going vehicle or used in conjunction with an internal combustion engine to form a hybrid drivetrain.

During the course of the project many areas with potential for further investigation also became apparent. Most notably these include:

- Noise immunity of high power motor controls and microcontroller circuitry – noise has the potential to render an otherwise perfect design totally useless or even dangerous.
- Regeneration with series wound DC motors – regeneration with series wound DC motors is uncommon, but could be practical with the use of variable brush timing and power electronics. This would yield efficiency gains and extend the distance the EV is able to travel before needing to recharge.
- Testing and refinement of high power motor control circuits – development of a process to efficiently and effectively test high power circuits while minimising the financial cost and time taken.



## **List of References**

Davis, S 2004, *Power-MOSFET Gate Drivers*, viewed 8 August 2005, <[http://www.elecdesign.com/Files/29/8415/8415\\_01.pdf](http://www.elecdesign.com/Files/29/8415/8415_01.pdf)>

Ebenhoech, O 1998, *Otmar's Regen Article*, viewed 8 August 2005, <<http://www.evcl.com/regen/>>

Hill, P 2004, *A Technical Guide to Building Fighting Robots*, viewed 22 July 2005, <<http://homepages.which.net/~paul.hills/>>

Linden, D (ed.) 1995, *Handbook of Batteries*, 2<sup>nd</sup> edn, McGraw-Hill, New York.

Mappus, S 2003, *Optimizing MOSFET Gate Drive Voltage*, Texas Instruments Incorporated, USA

Microchip 2003, *PIC16F87XA Data Sheet*, Microchip Technology Inc, USA

National Semiconductor 2003, *LM2907/LM2917 Frequency to Voltage Converter*, National Semiconductor Corporation, USA

Stadler, C 2005, *PICPgm – Free PIC Development Programmer for Windows*, viewed 20 April 2005, <<http://www.members.aon.at/electronics/pic/picpgm/index.html>>

Valentine, R (ed.) 1998, *Motor Control Electronics Handbook*, McGraw-Hill, New York.

## **Bibliography**

- AhFock, T, *ELE3805 Power Electronics Principles and Applications Study Book 1 & 2* 2004, University of Southern Queensland, Toowoomba.
- Beaty, H & Kirtley, J 1998, *Electric Motor Handbook*, McGraw-Hill, New York.
- Dorf, R (ed.) 1997, *The Electrical Engineering Handbook, 2<sup>nd</sup> edn*, CRC-Press, Boca Raton.
- ELE1801 Electrical Technology Study Book 1* 2002, University of Southern Queensland, Toowoomba.
- Fairchild Semiconductor 2003, *TL431/TL431A Programmable Shunt Regulator*, Fairchild Semiconductor Corporation, USA
- Maxim 2002, *76V High-Side Current-Sense Amplifiers with Voltage Output*, Maxim Integrated Products, USA
- Maxim 2000, *+5V Powered Multichannel RS-232 Drivers/Receivers*, Maxim Integrated Products, USA
- National Semiconductor 2005, *LM5100A Data Sheet*, National Semiconductor Corporation, USA
- National Semiconductor 1999, *LM2576 Series SIMPLE SWITCHER 3A Step-Down Voltage Regulator*, National Semiconductor Corporation, USA
- Turner, M 1996, *All You Need To Know About Fans*, viewed 7 October 2005, <[http://www.electronics-cooling.com/Resources/EC\\_Articles/MAY96/may96\\_01.htm](http://www.electronics-cooling.com/Resources/EC_Articles/MAY96/may96_01.htm)>

# Appendix A

## Project Specification

## ENG4111/ENG4112 Research Project

### Project Specification

**For:** Dean Thompson

**Topic:** Four-quadrant DC adjustable-speed drive

**Supervisor:** Dr. Tony Ahfock

**Project Aim:** To design, construct and test a 4-quadrant power electronic drive for a battery operated series DC motor.

**Programme:** Issue A, 17<sup>th</sup> March 2005

1. Select a suitable power electronic circuit for 4-quadrant operation
2. Carry out a conceptual design for the control of the drive including speed control and protection.
3. Select and design the hardware for design implementation
4. Write appropriate software for the drive
5. Construct and test the drive


*If time permits:*

6. Install the drive on a 48V electric vehicle
7. Test the drive at higher power levels

**AGREED:**

  
\_\_\_\_\_ (Student)

18 / 3 / 2005

  
\_\_\_\_\_ (Supervisor)

18 / 3 / 05 :

# Appendix B

## **PIC16F877A Data Sheet Extracts**

(Note: The complete set of data sheets used is located on the accompanying CD-ROM.)



# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F876A
- PIC16F874A
- PIC16F877A

### High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input  
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

### CMOS Technology:

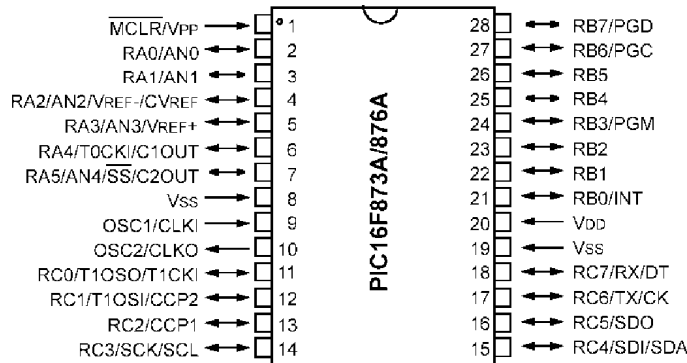
- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I <sup>2</sup> C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

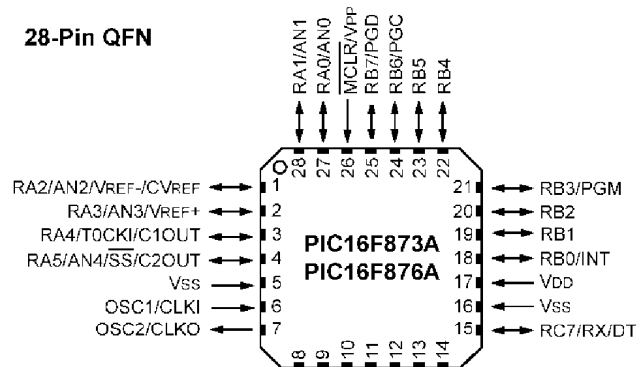
# PIC16F87XA

## Pin Diagrams

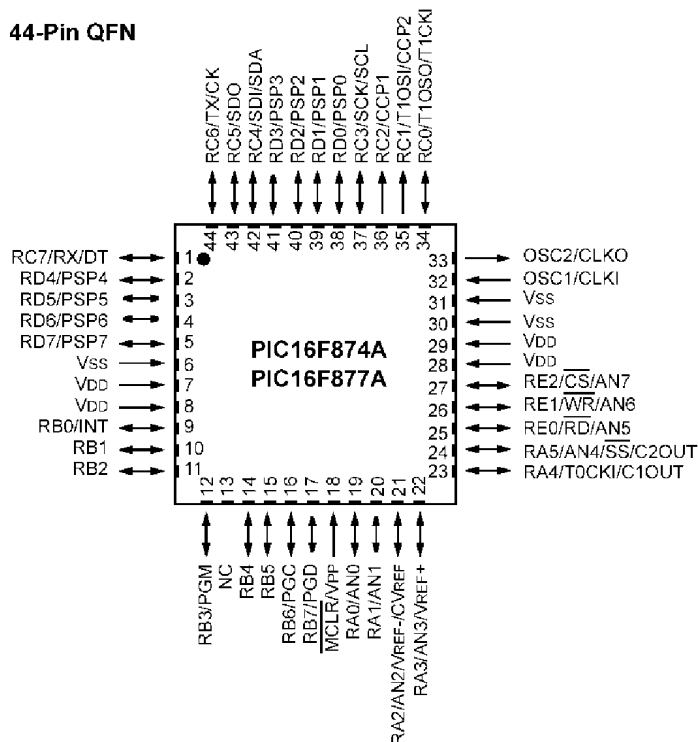
### 28-Pin PDIP, SOIC, SSOP



### 28-Pin QFN

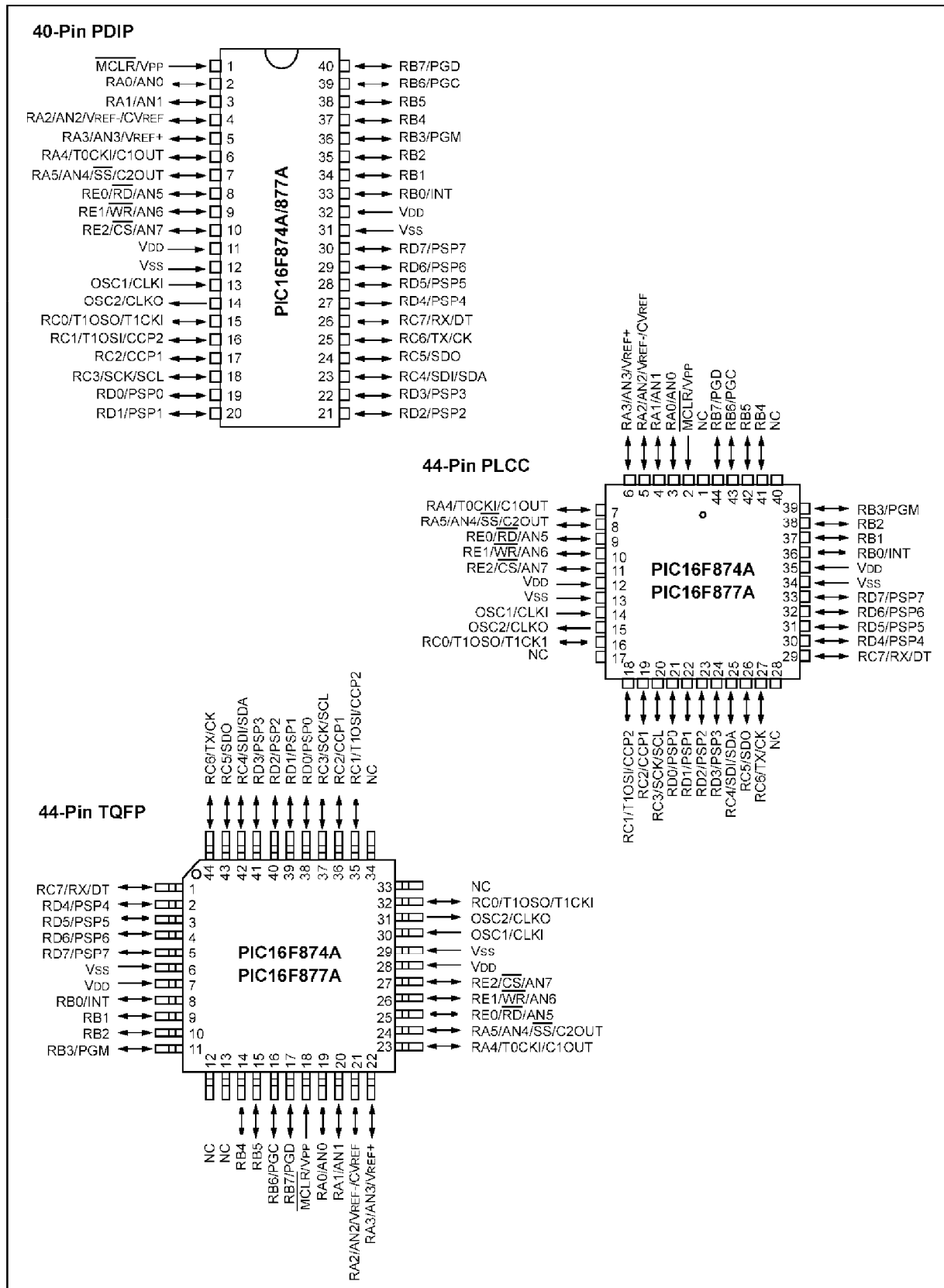


### 44-Pin QFN



# PIC16F87XA

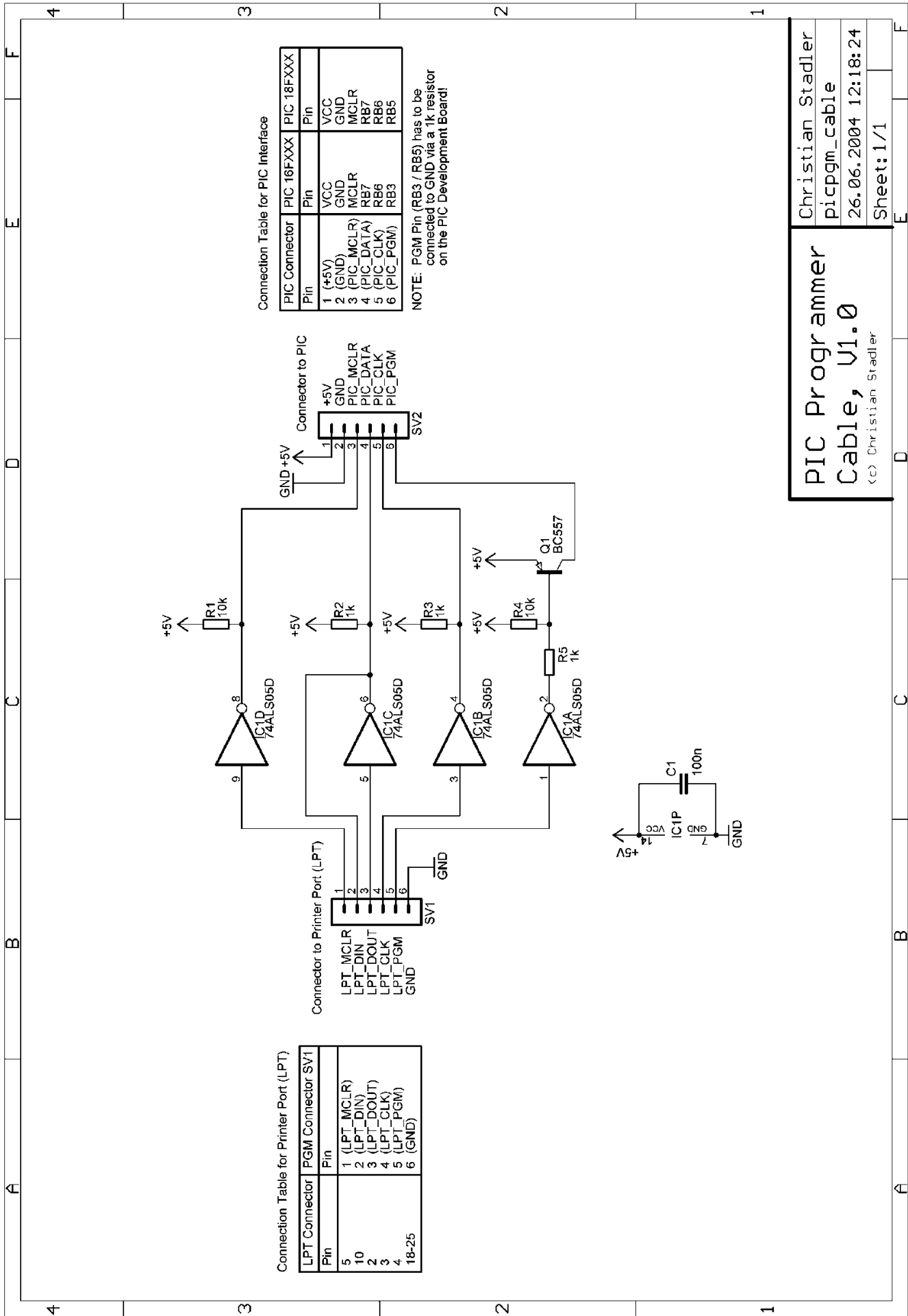
## Pin Diagrams (Continued)





## **Appendix C**

### **PIC Programmer Schematic Diagram**



Connection Table for PIC Interface

PIC Connector	PIC 16FXXX	PIC 18FXXX
1 (+5V)	VCC	VCC
2 (GND)	GND	GND
3 (PIC_MCLR)	MCLR	MCLR
4 (PIC_DATA)	RB7	RB7
5 (PIC_CLK)	RB6	RB6
6 (PIC_PGM)	RB3	RB5

NOTE: PGM Pin (RB3 / RB5) has to be connected to GND via a 1k resistor on the PIC Development Board!

Connection Table for Printer Port (LPT)

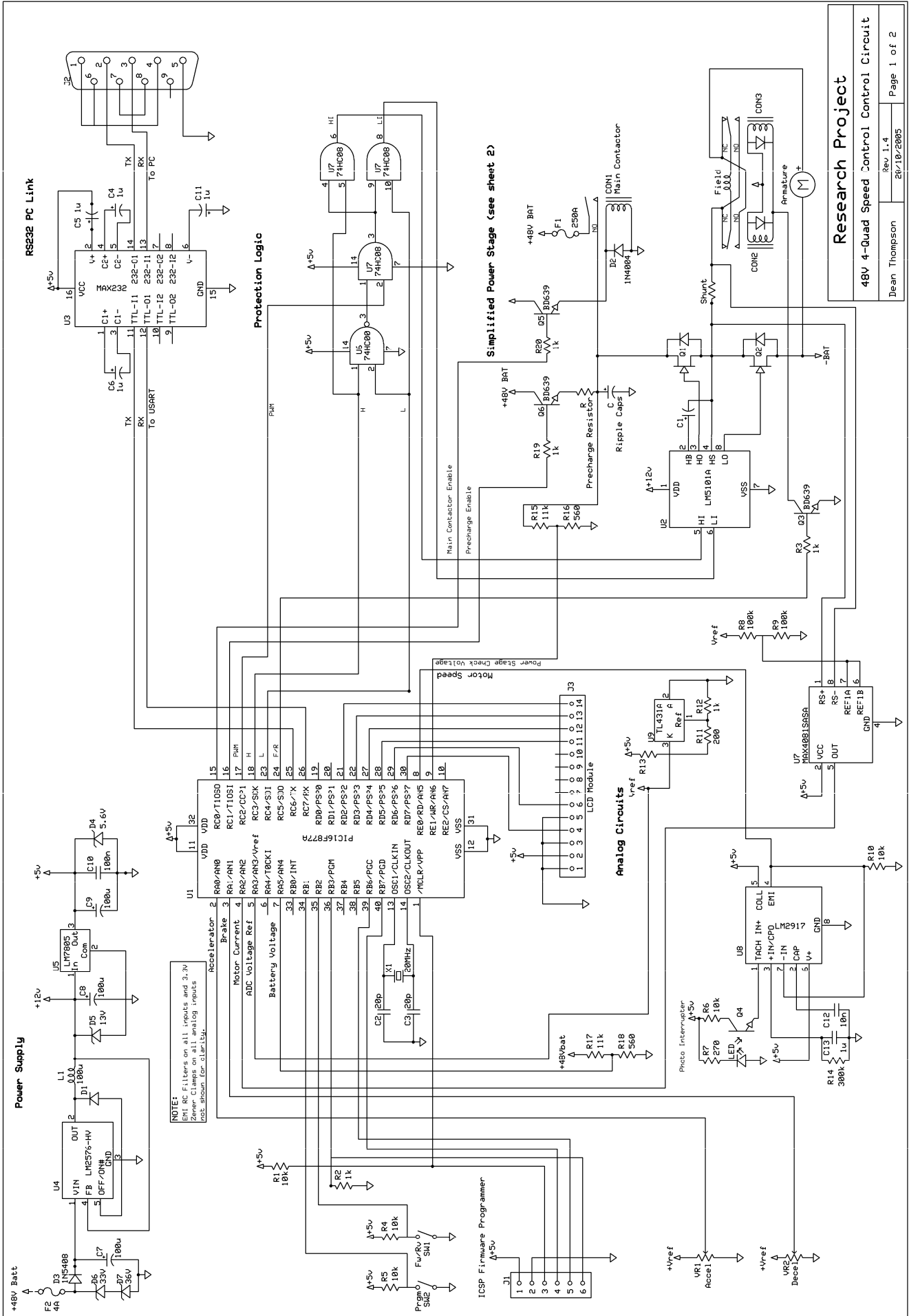
LPT Connector	PGM Connector SV1
5	1 (LPT_MCLR)
10	2 (LPT_BIN)
2	3 (LPT_DOUT)
3	4 (LPT_CLK)
4	5 (LPT_PGM)
18-25	6 (GND)

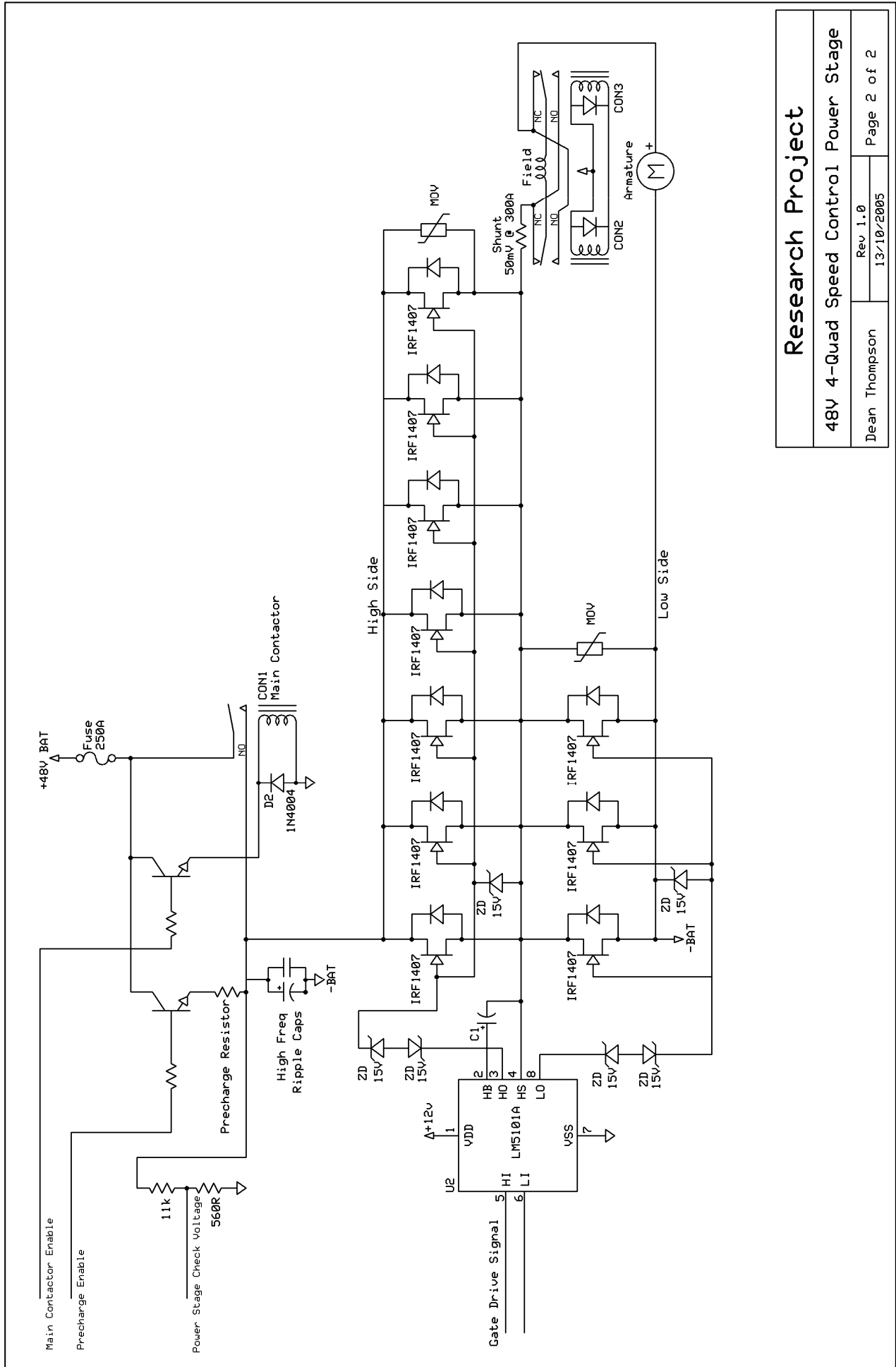
**PIC Programmer  
Cable, V1.0**  
(c) Christian Stadler

Christian Stadler  
picpgm\_cable  
26.06.2004 12:18:24  
Sheet: 1/1

# Appendix D

## Power Stage and Control Circuit Diagrams





# Research Project

48V 4-Quad Speed Control Power Stage

Rev 1.0  
13/10/2005

Dean Thompson

Page 2 of 2

# Appendix E

## **Microcontroller Software Code Listing**

(Note: The Mikrobasic source files are also located in the accompanying CD-ROM.)

```
'Firmware for 4 Quadrant Series DC Motor Control REV1
'(c) Dean Thompson 2005
'Firmware version 1 (27/9/2005)
```

```
program MotorControl
```

```
'Define variables and set defaults
```

```
'Misc variables
```

```
Dim data as Byte
```

```
Dim i as Integer
```

```
Dim direction as Byte
```

```
Dim temp as Word
```

```
Dim accel as Word
```

```
Dim decel as word
```

```
Dim currentlimit as Word
```

```
'Drive parameters
```

```
Dim maxbatv As Word      'Maximum allowable battery voltage
```

```
Dim minbatv As Word      'Minimum allowable battery voltage
```

```
Dim maxfwdspd As Word    'Maximum forward motor speed
```

```
Dim maxrevspd As Word    'Maximum reverse motor speed
```

```
Dim maxfwdmoti As Word   'Maximum forward motoring current
```

```
Dim maxrevmoti As Word   'Maximum reverse motoring current
```

```
Dim maxregeni As Word    'Maximum regeneration current
```

```
Dim shuntrating As Word  'Shunt rating (amps at 50mV)
```

```
'Program Subroutines
```

```
'Shutdown system
```

```
sub procedure Halt
```

```
'Stop PWM
```

```
Pwm_Stop
```

```
'Turn of power stage (including main contactor)
```

```
PORTC=0
```

```
'Display halt message
```

```
lcd_out(2, 1, "System Halted.      ")
```

```
'Stop program in an endless loop
```

```
while TRUE
```

```
wend
```

```
end sub
```

```
'Read EEPROM Parameters
```

```
sub procedure ReadParameters
```

```
data = Eeprom_Read(0)
```

```
maxbatv = (data << 8) + Eeprom_Read(1)
```

```
data = eeprom_read(2)
```

```
minbatv = (data << 8) + eeprom_read(3)
```

```
data = eeprom_read(4)
```

```
maxfwdspd = (data << 8) + eeprom_read(5)
```

```
data = eeprom_read(6)
```

```
maxrevspd = (data << 8) + eeprom_read(7)
```

```
data = eeprom_read(8)
```

```
maxfwdmoti = (data << 8) + eeprom_read(9)
```

```
data = eeprom_read(10)
```

```
maxrevmoti = (data << 8) + eeprom_read(11)
```

```
data = eeprom_read(12)
```

```
maxregeni = (data << 8) + eeprom_read(13)
```

```
data = eeprom_read(14)
```

```
shuntrating = (data << 8) + eeprom_read(15)
```

```
end sub
```

```
'Programming mode
```

```
sub procedure ProgrammingMode
```

```
lcd_cmd(lcd_clear)
```

```
lcd_out(1, 1, "Programming Mode")
```

```

lcd_out(2, 1, "Connect PC      ")
usart_init(2400) 'Setup USART for 2400 Baud, 1 stop bit
'Wait for response from PC
While usart_data_ready <> 1
Wend
data = Usart_Read
lcd_out(2, 1, "Sending data... ")
'Send out parameters
usart_write(255) 'Start byte
usart_write(hi(maxbatv))
usart_write(lo(maxbatv))
usart_write(hi(minbatv))
usart_write(lo(minbatv))
usart_write(hi(maxfwdspd))
usart_write(lo(maxfwdspd))
usart_write(hi(maxrevspd))
usart_write(lo(maxrevspd))
usart_write(hi(maxfwdmoti))
usart_write(lo(maxfwdmoti))
usart_write(hi(maxrevmoti))
usart_write(lo(maxrevmoti))
usart_write(hi(maxregeni))
usart_write(lo(maxregeni))
usart_write(hi(shuntrating))
usart_write(lo(shuntrating))
usart_write(255) 'End byte
'Wait for response from PC
lcd_out(1, 1, "Data sent.      ")
lcd_out(2, 1, "Waiting for PC. ")
While usart_data_ready <> 1
Wend
data = Usart_Read
'Read in settings from PC
lcd_out(2, 1, "Incomming data. ")
for i = 0 to 15
    while usart_data_ready <> 1
        Wend
        data = Usart_Read
        Eeprom_Write(i, data)
        usart_write(255)
    next i
    lcd_out(1, 1, "Data received. ")
    lcd_out(2, 1, "Programming Done")
    Delay_ms(2000)
    Halt() 'Stop and wait for reboot
end sub

'Check direction switch and change contactors if required
sub procedure CheckDirection
if PORTB.2 <> direction then
    direction=PORTB.2
    'Turn off PWM
    PWM_Change_Duty(0)
    PORTC.3=0
    PORTC.4=0
    'Wait till motor stops spinning
    lcd_out(1, 1, "Spinning down...")
    while adc_read(5) > 10
        wend
    'Change contactors
    if direction=0 then
        PORTC.5=1 'Set reversing contactors on
        lcd_out(1, 1, "Reverse!      ")

```



```

        currentlimit=maxrevmoti 'Buffer motor current limit for this
direction
    else
        PORTC.5=0 'Set reversing contactors off
        lcd_out(1, 1, "Forward!      ")
        currentlimit=maxfwdmoti 'Buffer motor current limit for this
direction
    end if
end if
end sub

'Precharge filter capacitors
sub procedure Precharge
'Switch on precharge transistor
PORTC.1=1
i=0
temp=1500 'Set to a high value so loop will start
'Wait until capacitors are within about 2V of battery voltage
while ADC_Read(7) < temp
    temp = ADC_Read(6) - 35 'Monitor voltage to ensure it rises
    i=i+1 'Increment timer
    if i=10 then 'If precharge takes longer than 10 seconds, there is
a fault
        lcd_out(1, 1, "Precharge fault!")
        Halt()
    end if
    Delay_ms(1000)
wend

'It's now safe to enable main contactor
PORTC.0=1 'Turn on main contactor
PORTC.1=0 'Turn off precharge
direction=PORTB.2 'Read current direction switch position
'Set reversing contactors if required
if direction=0 then
    PORTC.5=1 'Set reversing contactors on
    lcd_out(1, 1, "Reverse!      ")
    currentlimit=maxrevmoti 'Buffer motor current limit for this
direction
else
    PORTC.5=0 'Set reversing contactors off
    lcd_out(1, 1, "Forward!      ")
    currentlimit=maxfwdmoti 'Buffer motor current limit for this
direction
end if
end sub

'Main Program
main:
'Disable interrupts
intcon=0
'Initialise Ports
'Initialise LCD on port D
lcd_init(PORTD)
lcd_out(1, 1, "Motor Control")
lcd_out(2, 1, "(c)Dean Thompson")
delay_ms(500)
lcd_out(1, 1, "Self-Test...  ")
lcd_out(2, 1, "Please wait!   ")

'Initialise PWM on port C
PORTC=0
TRISC=%10000000 'Set input/output ports
T2CON=T2CON 'To fix bug in compiler

```

```

Pwm_Init(15000) 'Initialise PWM at 15KHz

'Initialise port B
PORTB=0
TRISB=%00000110 'Set input/output ports

'Initialise ADC on port A and E
ADCON1 = $82      'Configure analog inputs and Vref
TRISA=%11111111 'Port A is input
TRISE=%00000111 'Port E is input

'Load parameters from EEPROM
ReadParameters()

'Check validity of critical parameters and enter
'programming mode if invalid
if maxfwdmoti=$FFFF then
    ProgrammingMode()
end if
if maxfwdmoti=0 then
    ProgrammingMode()
end if
if minbatv=$FFFF then
    ProgrammingMode()
end if
if minbatv=0 then
    ProgrammingMode()
end if

'Check if programming switch is set
if PORTB.1=0 then
    ProgrammingMode()
end if

'Test for faults
'Check battery voltage, cancel startup if outside limits
if ADC_Read(4) > maxbatv then
    lcd_out(1, 1, "Bat voltage high")
    Halt()
end if
if ADC_Read(4) < minbatv then
    lcd_out(1, 1, "Bat voltage low ")
    Halt()
end if

'If pedals are pressed, cancel startup
if ADC_Read(0) > 10 then
    lcd_out(1, 1, "Accelerator on! ")
    Halt()
end if
if ADC_Read(1) > 10 then
    lcd_out(1, 1, "Decelerator on!")
    Halt()
end if

'Precharge filter capacitors and switch on power stage if successful
Precharge()

'Begin main program
PWM_Start
Pwm_Change_Duty(0) 'Start PWM and set duty cycle to 0

'Begin endless loop
while TRUE

```

```

'Clear watchdog timer
asm
CLRWDTC
end asm

'Do tests before switching on PWM to help avoid noise interference
'Check battery voltage
temp=ADC_Read(4)
'If voltage goes too high during regen (i.e. batteries fully charged)
then
'skip the cycle
if decel > 20 then 'If in regen mode
    if temp>954 then 'If battery voltage > 2.5V per cell then batteries
are charged
        lcd_out(1, 1, "Batteries Full! ")
        PORTC.3=0
        PORTC.4=0
        PWM_Change_Duty(0)
        goto skip          'Turn off PWM and skip the cycle
    end if
end if
'Otherwise if voltage is high or low for another reason, shut down the
system
if temp > maxbatv then
    lcd_out(1, 1, "Bat voltage high")
    Halt() 'Shut down
end if
if temp < minbatv then
    lcd_out(1, 1, "Bat voltage low ")
    Halt() 'Shut down
end if

'Check speed limits
temp=adc_read(5)
select case direction
    case 1 'Forward
        if temp >= maxfwdspd then
            lcd_out(1, 1, "Speed limit! ")
            PORTC.3=0
            PORTC.4=0
            PWM_Change_Duty(0) 'If speed limit reached then
            goto skip          'Turn off PWM and skip the cycle
        end if
    case 0 'Reverse
        if temp >= maxrevspd then
            lcd_out(1, 1, "Speed limit! ")
            PORTC.3=0
            PORTC.4=0
            PWM_Change_Duty(0) 'If speed limit reached then
            goto skip          'Turn off PWM and skip the cycle
        end if
    end select

'Check direction switch and change contactors if required
CheckDirection()

'Read pedal positions
accel=adc_read(0)
decel=adc_read(1)

'Enter regeneration mode if required (Brake takes preference over
accelerator)
if decel > 20 then 'Ignore slight pedal movement to combat noise

```

```

'Regen mode
lcd_out(2, 1, "Regeneration  ")
temp = adc_read(2)
'Current input swings from 1.5V for no current to 0V for max -ve
current
'If the current limit is already exceeded, don't turn on PWM
if temp <= maxregeni then
    PORTC.4=0
    PORTC.3=0
    goto skip
end if
'Select low side
PORTC.3=0
PORTC.4=1
'Set PWM duty cycle from pedal position
decel=decel>>2 'Convert to 8 bit number
PWM_Change_Duty(decel)
'Check for current limit during the PWM on time:
while TMR2 < PR2
    'Read motor current
    temp = adc_read(2)
    if temp <= maxregeni then '512=0 current, 0=max current
        'Deselect high and low sides to stop PWM
        PORTC.4=0
        PORTC.3=0
        break
    end if
    delay_cyc(10) 'Delay a few cycles so ADC read can work properly
wend
goto skip 'Restart main loop
end if

'Enter motoring mode if required
if accel > 20 then 'Ignore slight pedal movement to combat noise
    if decel < 20 then
        'Motoring mode
        lcd_out(2, 1, "Motoring  ")
        'If the current limit is already exceeded, don't turn on PWM
        'Current input swings from 1.5V for no current to 3V for max +ve
        current
        temp = adc_read(2)
        if temp >= currentlimit then '512=0 current, 1023=max current
            PORTC.4=0
            PORTC.3=0
            goto skip
        end if
        'Select high side
        PORTC.4=0
        PORTC.3=1
        'Set PWM duty cycle from pedal position
        accel=accel>>2 'Convert to 8 bit number
        PWM_Change_Duty(accel)
        'Check for current limit during the PWM on time:
        while TMR2 < PR2
            'Read motor current
            temp = adc_read(2)
            if temp >= currentlimit then
                'Deselect high and low sides to stop PWM
                PORTC.4=0
                PORTC.3=0
                break
            end if
            delay_cyc(10) 'Delay a few cycles so ADC read can work properly
        wend
    end if
end if

```

```
        goto skip 'Restart main loop
end if
end if

'This line should not be reached unless no pedals are pressed
PWM_Change_Duty(0) 'Reset PWM
lcd_out(2, 1, "Stopped      ")
'Deselect high and low sides
PORTC.4=0
PORTC.3=0
skip:
wend

'This part of the program should never be reached
lcd_out(1, 1, "Fatal Error!  ")
Halt()
End.
```

# Appendix F

## **PC Software Code Listing**

(Note: The Visual Basic source files are also located in the accompanying CD-ROM.)

```

'4 Quadrant Series DC Motor Control REV1
'PC software for programming drive parameters written in Visual Basic 6
'(c) Dean Thompson 2005
'Version 1.1 (28/9/2005)

'Define scaling multipliers
Dim multi(8) As Double

Private Sub ComPort_Click(Index As Integer)
'Change to selected COM port
NCD1.PortName = "COM" & Index
End Sub

Private Sub Connect_Click()
Dim a(17) As Long

'Define/Calculate scaling multipliers so that values are in the correct
'form for the microcontroller firmware.
'MaxBatV
multi(1) = 15.738
'MinBatV
multi(2) = 15.738
'MaxFwdSpd
multi(3) = 0.1705
'MaxRevSpd
multi(4) = 0.1705
'MaxFwdMotI
multi(5) = 511 / Val(ShuntRating.Text)
'MaxRevMotI.
multi(6) = 511 / Val(ShuntRating.Text)
'MaxRegenI.Text
multi(7) = 511 / Val(ShuntRating.Text)
'ShuntRating
multi(8) = 1

'Send or receive as appropriate
Select Case Connect.Caption
Case "CONNECT" 'Receive data
'Open Port and trigger transmission
Status.Caption = "Receiving..."
NCD1.OpenPort
NCD1.OutputByte 255 'start byte
'Wait for start byte
Do Until a(1) = 255
a(1) = NCD1.InputByte(10)
Loop
'Read in data
For t = 1 To 17
a(t) = NCD1.InputByte(10)
Next
Status.Caption = "Received OK"
Connect.Caption = "SEND"

'Close port, check for errors
NCD1.ClosePort
If a(17) <> 255 Then
MsgBox ("Checksum fail!")
Status.Caption = "RECEIVE ERROR! Start again."
Connect.Caption = "CONNECT"

Else
'Convert two 8 bit words to 16 bit integer form
MaxBatV.Text = Val((a(1)) * 256 + (a(2))) / multi(1)
MinBatV.Text = Val((a(3)) * 256 + (a(4))) / multi(2)

```

```

        MaxFwdSpd.Text = Val((a(5)) * 256 + (a(6))) / multi(3)
        MaxRevSpd.Text = Val((a(7)) * 256 + (a(8))) / multi(4)
        MaxFwdMotI.Text = (Val((a(9)) * 256 + (a(10))) - 512) / multi(5)
        MaxRevMotI.Text = (Val((a(11)) * 256 + (a(12))) - 512) /
multi(6)
        MaxRegenI.Text = (512 - Val((a(13)) * 256 + (a(14)))) / multi(7)
        ShuntRating.Text = Val((a(15)) * 256 + (a(16))) * multi(8)
    End If

Case "SEND" 'Transmit data
Connect.Caption = "CONNECT"
Status.Caption = "Sending..."
'Convert 16 bit integer to two 8 bit words
temp = Val(MaxBatV.Text) * multi(1)
a(1) = Int((temp And 65280) / 256)
a(2) = Int(temp And 255)
temp = Val(MinBatV.Text) * multi(2)
a(3) = Int((temp And 65280) / 256)
a(4) = Int(temp And 255)
temp = Val(MaxFwdSpd.Text) * multi(3)
a(5) = Int((temp And 65280) / 256)
a(6) = Int(temp And 255)
temp = Val(MaxRevSpd.Text) * multi(4)
a(7) = Int((temp And 65280) / 256)
a(8) = Int(temp And 255)
temp = (Val(MaxFwdMotI.Text) * multi(5)) + 512
a(9) = Int((temp And 65280) / 256)
a(10) = Int(temp And 255)
temp = (Val(MaxRevMotI.Text) * multi(6)) + 512
a(11) = Int((temp And 65280) / 256)
a(12) = Int(temp And 255)
temp = 512 - (Val(MaxRevMotI.Text) * multi(7))
a(13) = Int((temp And 65280) / 256)
a(14) = Int(temp And 255)
temp = Val(ShuntRating.Text) * multi(8)
a(15) = Int((temp And 65280) / 256)
a(16) = Int(temp And 255)

NCD1.OpenPort 'Open port
NCD1.OutputByte 255 'Signal beginning of transmission

'Send out data
For i = 1 To 16
    NCD1.OutputByte (a(i))
    Do Until NCD1.InputByte(10) = 255: Loop 'Wait until ready for
next byte
Next

Status.Caption = "Send OK!"
Connect.Caption = "CONNECT"

End Select
End Sub

Private Sub Exit_Click()
NCD1.ClosePort
Unload Programmer
End Sub

Private Sub Form_Load()
ComPort(1).Value = True 'Select COM1 as default
End Sub

```