DEVELOPMENT OF AN ENERGY MANAGEMENT SYSTEM FOR AN IN-HOUSE PARALLEL HYBRID ELECTRIC VEHICLE

BANCHAA INTHAKNU A/L PIN

This thesis is submitted as partial fulfillment of the requirements for the award of the
Bachelor of Electrical Engineering (Power Systems)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

JUNE, 2012

“All the trademark and copyrights use here in are property of their respective owner. References of information from other sources are quoted accordingly, otherwise the information presented in this report is solely work of the author”.

Signature : ______________________________

Name : BANCHAA INTHAKNU A/L PIN

Date : 21 JUNE 2012

To my beloved mother and father.

ACKNOWLEDGEMENT

I would like to thank my supervisor Mr. MohdRusllim bin Mohamed for giving me the opportunity to work on this intriguing project and Dr. Hamdan bin Daniyal, Dr. MudathirFunshoAkorede, Dr. Ahmed Mohamed Ahmed Haidar and Miss Nor Lailibinti Ismail for their constructive criticism during project evaluation, which helped me in making necessary improvements to my project.

I am deeply indebted to Mr. Shahrizal bin Saat whose constant guidance helped me completing my project.

I would also like to extend my gratitude to my family. The constant inspiration and guidance kept me focused and motivated. I am grateful to my father for giving me the life I ever dreamed. Words cannot ever be expressed enough to show how much grateful I am of my mother, whose unconditional love has been my greatest strength.

Finally, my greatest regards to the Almighty for bestowing upon me the courage to face the complexities of life and complete this project successfully.

ABSTRACT

Environmental as well as economic issues provide a compelling impetus to develop clean, efficient, and sustainable vehicles for urban transportation. Automobiles constitute an integral part of our everyday life, yet the exhaust emissions of conventional internal combustion engines (ICE) vehicles are to be blamed for the major source of urban pollution that causes the greenhouse effect leading to global warming. One of the ways to overcome this problem is the hybrid electric vehicle. The hybrid electric vehicle serves as a compromise for the environmental pollution problem and the limited range capability of today's purely electric vehicle. Hybrid electric vehicles have an electric motor as well as an internal combustion engine (ICE) to provide extended range and to ease down the pollution problem. The purpose of this study is to develop and model the energy management system of two energy sources in a hybrid electric vehicle which are the internal combustion engine and the electric motor. A small scale of a hybrid electric vehicle is constructed from a 1/10th size of a nitro-engine remote control car coupled with an electric motor. The energy management system is then developed by using microcontrollers and the power division between the two energy sources is controlled by the state of charge of the motor's battery. At the end of this project, the rpm of the motor is managed to be dependent on the state of charge of the battery and the lower the percentage of the battery state of charge, the lower the rpm of the electric motor and the higher the rpm of the engine, and vice versa.

ABSTRAK

Alam sekitar serta isu-isu ekonomi adalah antara faktor utama yang menyumbang kepada pengeluaran kenderaan yang lebih bersih, cekap, dan mapan. Kenderaan merupakan sebahagian daripada kehidupan seharian kita, namun pelepasan sisa ekzos enjin konvensional merupakan punca utama pencemaran udara di bandar yang menyebabkan kesan rumah hijau dan akhirnya membawa kepada pemanasan global. Salah satu cara untuk mengatasi masalah ini adalah kenderaan elektrik hibrid. Kenderaan elektrik hybrid berfungsi sebagai kompromi kepada masalah pencemaran alam sekitar. Tujuan kajian ini adalah untuk mereka model sistem pengurusan tenaga kenderaan hybrid elektrik yang bersaiz kecil. Model bersaiz kecil kenderaan elektrik hybrid dibina daripada saiz 1/10 sebuah kereta kawalan jauh nitro-enjin ditambah dengan sebuah motor elektrik. Sistem pengurusan tenaga kemudiannya dibangunkan dengan menggunakan mikropengawal dan pembahagian kuasa di antara kedua-dua sumber tenaga dikawal oleh jumlah caj bateri. Pengawalan kelajuan motor bergantung kepada peratusan caj dalam bateri dan semakin rendah peratusan caj bateri, semakin rendah kelajuan motor elektrik dan semakin tinggi kelajuan enjin, begitu juga sebaliknya.

TABLE OF CONTENTS

2 LITERATURE REVIEW

3 METHODOLOGY

4 RESULT AND DISCUSSION

5 CONCLUSION AND RECOMMENDATION

APPENDIXES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PWM	-	Pulse Width Modulation
HEV	-	Hybrid Electric Vehicle
EV	-	Electric Vehicle
PIC	-	Peripheral Interface Controller

CHAPTER 1

INTRODUCTION

1.1 Background

A hybrid electric vehicle (HEV) is a vehicle which at least one of the energy sources, stores, or converters can deliver electric energy. A hybrid road vehicle is one in which the propulsion energy during specified operational missions is available from two or more kinds or types of energy stores, sources or converters, of which at least one store or converter must be on board.

The HEV serves as a compromise for the environmental pollution problem and the limited range capability of today's purely electric vehicle. HEVs has an electric motor as well as an internal combustion engine (ICE) to provide extended range and to ease down the pollution problem.

1.2 Problem Statement

The dependence on oil as the source of energy for passenger vehicles has economics and political implications, and the crisis will inevitably become acute as the oil reserve of the world diminishes. The number of automobiles being introduced on the road every year is only adding to the pollution problem. There is also an economic factor inherent in the poor energy conversion efficiency of combustion engines.

Electric vehicles enabled by high-efficiency electric motors and controllers and powered by alternative energy sources provide the means for a clean, efficient and environmentally friendly urban transportation system.

1.3 Objectives

The objectives of this project are:

i. To create the energy management system of an in-house parallel hybrid electric vehicle.

ii. To develop electronic systems to demonstrate the power division.

1.4 Scope of Project

This project is focused on developing electronics components to manage the two energy sources of a parallel hybrid electric vehicle. A Radio-Controlled Nitro Car is used as an in-house model of a vehicle.

The electronics systems are designed to divide and manage the power from both sources by depending on the state-of-charge of the 9.6V battery of the motor. For example, when the state-of-charge of the battery is determined to be at 100%, the power supplied by the battery will be 75% of the total resultant speed of the vehicle.

1.5 Chapter Outlines

This thesis is composed of five different chapters inclusive of this chapter. Each and every chapter’s content can be indited as follows:

Chapter 2 provides a literature review, background, previous research done by other researchers in the same area and relevant issues pertaining to the development of the energy management system of parallel hybrid electric vehicles. This includes a brief history on the beginning of the first ever hybrid electric vehicle that dated back in the year of 1834. The emergence and failure of electric vehicles which paved up the way for hybrid electric vehicles in the automobile industry, architecture of hybrid electric vehicles and the advantages of parallel hybrid electric vehicles over series hybrid electric vehicles followed afterwards.

Chapter 3 describes a vast description of the research methodology in this study. This chapter begins with description of procedures undergone in order for this study to be completed which consists of four phases starting from the literature review to the testing and running the hardware developed. The construction of programs used in this study, which is the dc motor speed controller program and servo motor controller program are overviewed and extensively explained.

Chapter 4 presents on the result and discussion on the testing and running the fully developed system. In this chapter, the aim is to match the expected outcomes of this study to the results obtained. The results obtained from the testing done do not match and conformed to the expected outcomes. Failure mode and effects analysis of the system is widely elaborated in this chapter to identify the potential causes of the failure of the system.

Chapter 5 provides a general conclusion about this study. Recommendations for future development of this project are also suggested.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents the fundamental theory about hybrid electric vehicle and type of hybrid electric vehicles.. This chapter is also focusing on the past researches which are related with this project.

2.2 A Brief History of Hybrid Electric Vehicles (HEVs)

The invention of electric vehicles began in the year of 1834, but the world's first hybrid electric vehicle was not built until the year 1898, by Dr. Ferdinand Porsche. He used an internal combustion engine to turn a generator that provided power to electric motors that were specifically located in the wheel hubs [1]. In 1903, another variation of HEV was built by the Krieger Company. This hybrid vehicle used a gasoline engine, in addition to an electric motor with a battery pack, to provide propulsion power needed to drive the vehicle [2]. Both kinds of hybrid vehicles are much alike to the ones that we have today.

Moreover, in the 1900s, Pieper, a Belgian car manufacturer, initiated a 3.5 hp ‘Voiturette’ or a small automobile, wherein a small gasoline engine was coupled to an electric motor. The electric motor in the vehicle was used as a generator for charging batteries when the car was driven under a sustained velocity, or coasting. Once the car is picking up its velocity, the electric motor which is mounted in parallel with the gas engine, aided the gasoline engine to propel the vehicle. By the year of 1905, a US engineer named H. Piper, filed a copyright for a petrol-electric hybrid vehicle. His idea was to employ the electric motor to provide some means of aid to an internal combustion engine. Both hybrid designs show some similarities to the modern-day parallel hybrid electric vehicles [3].

Collectively, HEVs and EVs, vanished from the market by the year of 1930 and the electric car fabricators are all ceased to continue their car-making business. There were many ends to why HEVs and EVs faded away from the market. In contrast to the conventional gasoline-powered vehicles, EVs and HEVs were way more costly and pricy because of the added large battery packs. EVs and HEVs were also short of performance required in term of fewer power produced than gasoline cars, owing to the restricted amount of power from the onboard battery. Another major restriction of EVs and HEVs would be the limited span amid each charge and the onboard batteries consumed many hours to restore the batteries to their full charge in order for the vehicles to be back on the road again [4].

In the year of 1997, the first modern hybrid electric car, the Toyota Prius, was presented and sold in Japan [15]. This vehicle, together with Honda's Insight and Civic HEVs, has been made obtainable globally since 2000. This initial expansion and elaboration of HEVs indicated a fundamental advancement in the automotive world and the sorts of cars being proposed to the public: automobiles that merged the benefits of both the battery-electric vehicles and conventional internal combustion engine vehicles. As this report is written, there are beyond 40 models of HEVs available in the automotive arena from more than 10 main car companies [5].

2.3 The Emergence and Failure of Electric Vehicles (EVs) in the 1990s

During 1990s, immense smog and pollution problem appeared in some portions of the world. Taking advantage of this situation, all major car manufacturers begin to produce electric cars. In just short years' time, consumers were looking at 10 production electric vehicles at a particular time. Regrettably, the arcade of electric automobiles diminished in the late 1990s. One of the major explanations to the fall of electric vehicles would be the limitations that electric vehicles have [11]. These concerned to the inadequacy of range (most electric vehicles delivered 100 – 160 kilometers, rivaled to 480 or more kilometers from gasoline-powered vehicles) ; lengthy charging period (eight or more hours) ; high price (40 % more expensive than gasoline-powered cars) ; and reduced amount of goods space in most of the EVs available. The subsequent cause that backed the failure of EVs is that gasoline or fuel costs way less in 1990s. The operation charge (fuel fee) of automobiles is immaterial in contrast to the financing that the owner of an electric vehicle has to commit in acquiring an electric vehicle [18]. Also, one of the major backers to the collapse of EVs in 1990s was the battery technology. In the 1990s, lead acid batteries were the option back then [17]. The batteries were huge and weighty, and required long time to charge. The final cause of the vanishing of EVs from the industry in 1990s would be the amenity provided. There was limited infrastructure for charging batteries.

As we attempt for a solution towards affording sustainable transportation to consumers, lessons learnt will keep the history from recurring itself. In the present setting of HEV, several obstructions have to be cleared in order to succeed. The primary one has to be the vital technology which is the batteries, power electronics, and electric motors. Explicitly, the starved of breakthroughs in batteries technology and with the cost of fuel striving to be at low levels, significant hurdles will be foreseen for huge scale distribution of HEVs. The subsequent obstacle would be the price. HEVs are priced at a higher level than the conventional internal combustion engine vehicles. Component and system costs have to be lowered down to a satisfying level [18]. Facility matter has to be addressed too. These requirements are to be completed for the massive distribution of plug-in hybrid electric vehicles (PHEVs) and enlarged renewable energy generation, and also for express and expedient charging of grid PHEVs. And the last approach would be the method by which will be the system of HEVs will be built of. A cohesive approach that merges high-efficiency engine, vehicle security, and smarter roadways will ultimately assist in the act of shaping sustainable and maintainable future for private transportation [7].

2.4 Architecture of HEVs

A hybrid electric vehicle is defined by the combination of a conventional internal combustion engine (ICE) – powered vehicle and an electric motor/generator for propulsion and thrust. From a power flow viewpoint, the ICE and the electric motor can be associated whether in series or in parallel with each other. The HEV is a series hybrid by the mean of connecting the ICE in series with the electric motor, and in this connection, only the electric motor is offering mechanical power to the wheels. When the ICE is connected in parallel with the electric motor, the HEV is a parallel hybrid in which both the electric motor and the ICE can provide mechanical power to the wheels [19].

Fluid fuel is still the main source of energy in a HEV. The ICE is the focal power switcher that affords all the energy for the vehicle. The electric motor increases the system competency and reduces the fuel ingestion by the act of recuperating the kinetic energy during regenerative braking and enhancing the operation of the ICE during usual driving day by regulating the engine torque and speed. The ICE affords the vehicle with a prolonged driving range therefore eliminating the drawbacks of a pure electric vehicle [20].

In addition to the liquid fuel available on a plug – in hybrid electric vehicle (PHEV) , there is also stored electricity in the battery, which can be charged from the electric grid. Consequently, fuel ingestion can be further lessened.

In a series HEV or PHEV, a generator is driven by the ICE (referred to as the I/G set) . The ICE converts energy in fluid fuel to mechanical energy and the generator switches the mechanical energy of the engine output to electricity. An electric motor will impel the vehicle by using the electricity produced by the I/G set. The electric motor is also used to seize the kinetic energy during braking. A battery is placed in the middle of the generator and the electric motor to buffer the electric energy between the I/G set and the motor [21].

In a parallel HEV or PHEV, both the ICE and the electric motor are attached to the final drive shaft through a mechanical coupling mechanism. Parallel configuration concedes both the ICE and the electric motor to provide propulsion power to the vehicle either in joined mode, or isolatedly. The usage of the electric motor can also be varied for regenerative braking and for interning the extra energy from the ICE during freewheeling [8].

2.4.1 Series HEVs

Figure 2.1 shows the configuration of a series HEV. In this type of HEV configuration, the ICE is the main energy converter that switches the original energy in fuel to mechanical power. The generator then converts the mechanical output of the ICE into electricity. The vehicle is then propelled by the mean of the electric motor using electricity generated by the generator or electricity stored in the battery. The source of electricity for the electric motor can come from the engine, or from the battery, or both. The engine velocity can be governed independently of vehicle speed since the engine is decoupled from the wheels. This feature does not only make the control of the engine easier, but most significantly, permits the operation of the engine at its peak speed to attain the best fuel economy. Moreover, it offers suppleness in locating the engine on the vehicle. The old – fashioned mechanical transmission is unnecessary in a series HEV. Grounded on the vehicle working conditions, the component of propulsion of a series HEV can function with diverse combinations :

- Battery alone: When the energy in the battery is adequate, and the requirement of power from the vehicle is low, the I/G set is turned off, with the battery being the only source of power.
- Combined power: When the power requirement is high, the I/G set is switched on and the power to the electric motor is provided by the battery as well.
- Engine alone: The I/G set is turned on when the vehicle requires moderately high power and also during freeway cruising. The battery charge and discharge operation are brought to a halt. This condition is due to the event that the state of charge (SOC) of the battery is already at a satisfactory level but the power request of the vehicle inhibits the engine from turning off, or it may not be effective to turn the engine off.
- Power split: When the I/G set is turned on, the I/G set optimum power is greater than the vehicle power requirement, and the SOC of the battery is low, then a small allotment of the I/G power is utilized for battery charging.
- Stationary charging: The I/G set provides electric power to charge the battery without the vehicle being driven.
- Regenerative braking: The electric motor performs as a generator in order to convert the kinetic energy of the vehicle into electricity and charge the battery [11].

The configuration of a series HEV can be in the same way of that conventional vehicles' configuration, and that is the electric motor is placed in front of the engine as shown in Figure 2.1.

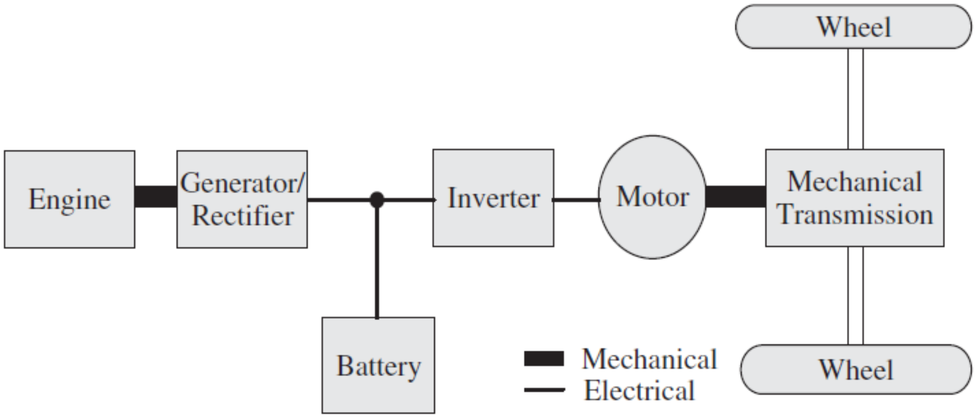


Figure 2.1 The architecture of a series HEV [11]

2.4.2 Parallel HEVs

Figure 2 shows the parallel hybrid electric vehicle configuration. In this arrangement, both of the ICE and the electric motor can provide propulsion power to the vehicle. The ICE and the electric motor are coupled to the final drive through a mechanism. Power can be provided to the final drive from the ICE and the motor, either in joined mode, or each isolatedly. The kinetic energy can be restored during braking and a measure of the ICE power can be absorbed by converting the electric motor as a generator. There are only two propulsion devices required in the parallel hybrid electric vehicle and they are the ICE and the electric motor, which can be used in the following mode :

- Motor-alone mode: If the battery contained sufficient energy, and the vehicle requires only low power, the engine is then switched off, and the mean of propulsion of the vehicle would be from the motor and the battery only.
- Combined power mode: When high amount of power is requested, the engine is turned on and the power to the wheels is also supplied by the electric motor.
- Engine-alone mode: During highway cruising and at mildly high power requirement, the engine offers all the propulsion power needed to drive the wheels. The electric motor continues to be indolent. This is because the battery SOC is already at an optimum level but the power requirement of the vehicle averts the engine from being turned off, or it may be inefficient to turn the engine off.
- Power split mode: When the engine is functioning, but the vehicle requires low power and the SOC of the battery is low, a fraction of the engine power is switched to electricity by the electric motor in order to charge the battery.
- Stationary charging mode: The motor is operated as a generator to charge the battery without the vehicle being propelled.
- Regenerative braking mode: The vehicle’s kinetic energy is converted into electric energy, and stored in the battery by running the electric motor as a generator [11].

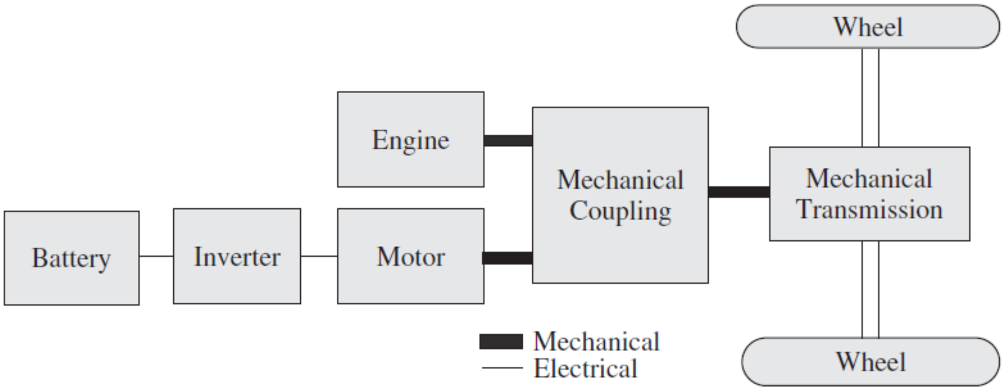


Figure 2.2 The architecture of a parallel HEV [11]

2.4.3 The Advantages of Parallel HEVs Over Series HEVs

In the series hybrid, a generator is powered by an internal combustion engine which charges the battery or provides power right to the propulsion circuit and thus lowers the power request on the battery. In a parallel hybrid, the mechanical power is directly delivered to the drivetrain by the internal combustion engine, eliminating the generator. With this sort of configuration, the propulsion power may be provided by the battery-electric system or the internal combustion engine, or they may be utilized concurrently for optimum power. In term of efficiency, the favor is in the parallel hybrid configuration. This point is backed up by the fact that parallel hybrids provide internal combustion engine – power (mechanical power) directly to the drive wheels, instead of having to undergo a conversion process of mechanical power to electricity.

Each and every time one form of power is converted to another, losses occur. An internal combustion engine runs a generator to produce electricity in the series hybrid electric vehicle. The mechanical to electrical power conversion causes loss of about 20 percent [11]. The electricity may be distributed to a battery or to a motor in order to provide a motive power. If the electricity is distributed to a motor, the electrical energy is converted into mechanical power by the motor, but the result of the conversion process would be a loss in power of about 20 percent. If electrical energy is to be delivered to a battery, there will be an additional 20 to 30 percent of energy absent in the conversion process (putting the charge into the battery and then taking it out again). As the internal combustion engine yields mechanical power in the first place, the level of efficiency will be a lot higher if the engine is working in its original form rather than performing a double conversion – from one form to another form and then back to its original form [9].

2.5 Control Strategies and Systems of HEVs

One of the most important research and development topic for hybrid electric vehicle (HEVs) is a pioneering hybrid drivetrain, whose factors modification is a must in order to achieve certain desired performance of the hybrid vehicle [4]. The main components that are contained in a hybrid powertrain are like electric motors, power electronic converters, energy storage device such as batteries and ultracapacitors, and complex controllers in accumulation to such orthodox components as internal combustion engine (ICE) , transmissions, clutch, drive shafts, differentials, and so on. Consequently, a hybrid powertrain is much more complex compared to a conventional powertrain. [12]

In order to determine hybrid vehicle component sizes, a parametric scheme method can be utilized. The design variables comprise the power rating of the fuel converter, motor controller, a number of battery cells, the final drive ratio, and control strategy factors. One of the advantages that parametric design has is that it produces better hybrid vehicle with improved performance compared to the standard one. It is impossible that the whole design is to be at its finest state, but it can be utilized as an estimated design for additional optimization in order to make the most out of the fuel economy and reduce emissions, weight and cost. Simultaneously, the demands of vehicle operation have to be gratified [6].

A control plan which is situated in the vehicle dominant controller, is outlined as a set of rules, or algorithm, which is a principle normalizing the procedure of the vehicle drivetrain. Commonly, the algorithm enters the dimensions of the vehicle operating conditions such as speed or acceleration, demanded torque by the driver, existing road type or traffic information, in – advance solutions, and also the information provided by the Global Positioning System (GPS) . The outcomes of a control approach are decisions on whether to switch certain components ON or OFF or ruling local component controllers so as to modify their operation region. For an instance, the internal combustion engine can be ruled to operate near to its most competent efficiency curve, utilizing an electrical motor which acts like a buffer to balance the load.

The main purposes of managing the hybrid drivetrain energy are satisfying the drivers’ requirement for the traction power, maintaining the battery charge and maximization of drivetrain efficiency, fuel consumption, and emanations. Even so, some of these goals, such as emission reduction and efficiency optimization, are opposing parameters, and an effective control method should fulfill a tradeoff among them. [10]

For the purpose of developing a small scale functional model of an energy management of a parallel hybrid electric vehicle, microcontrollers are used as primary controllers to divide the power between the two energy sources of a small scale hybrid electric vehicle depicted in a 1/10th radio-controlled car.This report overviews the main proposed approach to the energy management problem in HEVs for this project, which is the fuzzy rule – based control strategy.

2.6 Radio-controlled (R/C) Car

Radio-controlled (or R/C) cars are self-powered model cars or trucks that can be controlled from a distance using a specialized transmitter. The term "R/C" has been used to mean both "remote-controlled" and "radio-controlled", where "remote-controlled" includes vehicles that are connected to their controller by a wire, but common use of "R/C" today usually refers to vehicles controlled by a radio-frequency link [22].

Cars are powered by various sources. Electric models are powered by small but powerful electric motors and rechargeable nickel-cadmium, nickel metal hydride, or lithium polymer cells. There are also brushed or brushless electric motors. Most fuel-powered models use glow plug engines, small internal combustion engines fueled by a special mixture of nitromethane, methanol, and oil (in most cases a blend of castor oil and synthetic oil). These are referred to as "nitro" cars [23]. Recently, exceptionally large models have been introduced that are powered by small gasoline engines, similar to string trimmer motors, which use a mix of oil and gasoline. Electric cars are generally considered easier for the novice to work with compared to fuel-driven models, but can be equally as complex at the higher budget and skill levels [24].

In both of these categories, both on-road and off-road vehicles are available. Off-road models, which are built with fully functional off-road suspensions, and a wide tire selection, can be used on various types of terrain. On-road cars, with a much less robust suspension, are strictly limited to smooth, paved surfaces. In the past decade, advances in "on-road" vehicles have made their suspension as adjustable as many full scale race cars, today.

2.6.1 Principles of Operation of R/C Cars

Radio-controlled cars use a common set of components for their control and operation. All cars require a transmitter, which has the joysticks for control, or in pistol grip form, a trigger for throttle and a wheel for turning, and a receiver which sits inside the car. The receiver changes the radio signal broadcast from the transmitter into suitable electrical control signals for the other components of the control system. Most radio systems utilize amplitude modulation for the radio signal and encode the control positions with pulse width modulation [25]. Upgraded radio systems are available that use the more robust frequency modulation and pulse code modulation. Recently however, 2.4Ghz frequency radios have become the standard for hobby-grade R/C cars. The radio is wired up to either electronic speed controls or servomechanisms (shortened to "servo" in common usage) which perform actions such as throttle control, braking, steering, and on some cars, engaging either forward or reverse gears. Electronic speed controls and servos are commanded by the receiver through pulse width modulation; pulse duration sets either the amount of current that an electronic speed control allows to flow into the electric motor or sets the angle of the servo. On these models the servo is attached to at least the steering mechanism; rotation of the servo is mechanically changed into a force which steers the wheels on the model, generally through adjustable turnbuckle linkages. Servo savers are integrated into all steering linkages and some nitro throttle linkages. A servo saver is a flexible link between the servo and its linkage that protects the servo's internal gears from damage during impacts or stress [22].

2.6.2 Nitro-powered R/C Cars

Nitro-methane fuel powered models utilize a single servo for throttle and braking control; rotation of the servo in one direction will cause the throttle on the carburetor to open, providing more air and fuel mixture to the internal combustion engine. Rotation of the servo in the other direction causes torque to be applied to a linkage and cam which causes friction with the braking material. The brake is commonly located on the driveshaft or spur gear in some cases and applies stopping power only to the driven wheels. Some models will also use an additional servo to control a transmission box, enabling the vehicle to drive in reverse [22].

Fuel engine sizes most often range between 0.12-0.35 cubic inches. This is due to restrictions by the main sanctioning bodies for radio-controlled racing. Many "outlaw" engines are manufactured larger than these, mainly intended for vehicles which will not be used in sanctioned races and therefore do not need to comply with these regulations. Engine size is related to the class of car; 1/10 scale on and off road vehicles usually are equipped with 0.12-0.18 cubic inch engines, with 1/8 scale vehicles using 0.21-0.32 cubic inch engines. Fuel-powered engines allow model cars to reach moderate speeds unmodified [26]. Maximum power is generally achieved at medium to high speeds, and a slightly slower throttle response than electrically powered vehicles is to be expected due to clutching and lack of torque. Electric motors effectively produce instantaneous torque, whereas nitro engines, like full-sized gasoline engines, take time for the engine to spool up and for the clutch to engage. Nitro- (and fuel) powered cars may be refueled and returned to action in a few seconds, as opposed to electrics needing to remove the body shell and battery fasteners to replace a discharged battery. Nitro cars are cooled some by air; some by the oil mixed in with the fuel and may be run continuously with no need to take breaks for cooling down assuming they are properly tuned [24].

Nitro-powered cars operate like full-sized fuel vehicles more than their electric counterparts do, making use of a two stroke engine rather than an electric motor. The sound of the engine and generally higher stock top speeds are main selling points to nitro enthusiasts. However, their exhaust contains unburned oil, which usually ends up coating the chassis. This, in turn, requires more cleaning than an electric-powered equivalent [23]. Cleaning is usually achieved by the use of compressed air nozzles and solvents (such as denatured alcohol). Tuning a fuel-powered vehicle requires learning to maintain optimum performance and fuel economy, and to minimize engine wear and overheating, even in ready-to-run vehicles. Running a nitro-fuel motor without tuning or tuning improperly can hurt performance in rich conditions, and cause severe damage in lean conditions [24].

Because of higher stock performance and their ability to be driven for longer periods of time, mechanical wear in nitro vehicles is generally greater than in electric vehicles. In addition, the increased speed and weight of fuel-powered vehicles generally lead to higher speed collisions, causing greater damage to the collided vehicles, and a greater degree of safety concerns needs to be taken into account. Maintenance such as cleaning of the air filter and general chassis cleaning, replacement of worn clutch parts, proper after-run lubrication (necessary for storage) and maintenance of other motor-related items such as glow plug replacement makes for a more frustrating experience for first time RC users. In addition, nitro motors typically require rebuilding or replacement after 2-8 gallons of fuel run through them, due to loss of compression, which can be accelerated by poor tuning and overheating. It is also possible to seriously damage the engines by over-revving them with no load or ingestion of dirt into the carburetor. As such, nitro-powered vehicles are by nature expensive to maintain [24].

2.7 Direct-current (DC) Motor

A DC motor is an electric motor that runs on direct current (DC) electricity. DC motors were used to run machinery, often eliminating the need for a local steam engine or internal combustion engine. DC motors can operate directly from rechargeable batteries, providing the motive power for the first electric vehicles. Today DC motors are still found in applications as small as toys and disk drives, or in large sizes to operate steel rolling mills and paper machines. Modern DC motors are nearly always operated in conjunction with power electronic devices.

The torque in electric motors is produced by utilizing one of two basic principles of electromagnetic theory; by Lorentz force principle, where torque is produced by the mutual interaction of two orthogonal magnetomotive forces (mmf);

and by reluctance principle, where the rotor produces torque while moving toward the minimum reluctance position in a varying reluctance path.

The DC motors have two sets of windings, one in the rotor and the other in the stator, which establish the two fluxes; hence, the mmfs that interact with each other to produce the torque. The winding in the rotor is called the armature winding, while the winding in the stationary part of the machine is called the field winding. The armature and the field windings are supplied with DC currents[11].

Positive attributes of DC machines are as follows:

- Ease of control due to linearity
- Capability for independent torque and flux control
- Established manufacturing technology

2.7.1 Brushless DC Motors

Brushless DC motors use a rotating permanent magnet or soft magnetic core in the rotor, and stationary electrical magnets on the motor housing. A motor controller converts DC to AC. This design is simpler than that of brushed motors because it eliminates the complication of transferring power from outside the motor to the spinning rotor. Advantages of brushless motors include long life span, little or no maintenance, and high efficiency. Disadvantages include high initial cost, and more complicated motor speed controllers. Some such brushless motors are sometimes referred to as "synchronous motors" although they have no external power supply to be synchronized with, as would be the case with normal AC synchronous motors [11].

2.7.2 Brushed DC Motors

The brushed DC electric motor generates torque directly from DC power supplied to the motor by using internal commutation, stationary magnets (permanent or electromagnets), and rotating electrical magnets.

Like all electric motors or generators, torque is produced by the principle of Lorentz force, which states that any current-carrying conductor placed within an external magnetic field experiences a torque or force known as Lorentz force. Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed. Disadvantages are high maintenance and low life-span for high intensity uses. Maintenance involves regularly replacing the brushes and springs which carry the electric current, as well as cleaning or replacing the commutator. These components are necessary for transferring electrical power from outside the motor to the spinning wire windings of the rotor inside the motor. For the simplicity of this project, a brushed 9.6 V DC motor is chosen as a small scale representation of a life size electric motor in a hybrid electric vehicle [11].

2.8 Servo Motor

Servo refers to an error sensing feedback control which is used to correct the performance of a system. Servo or RC Servo Motors are DC motors equipped with a servo mechanism for precise control of angular position. The RC servo motors usually have a rotation limit from 90° to 180°. Some servos also have rotation limit of 360° or more. But servos do not rotate continually. Their rotation is restricted in between the fixed angles.The Servos are used for precision positioning. They are used in robotic arms and legs, sensor scanners and in RC toys like RC helicopter, airplanes and cars [27].

In order to accomplish a servo function, instantaneous positioning information of the output shaft is fed back to the control circuit using a transducer. A simplest way of doing this is by attaching a potentiometer to the output shaft or somewhere in the gear train. The control electronics compares the feedback signal (which contains the current position of the shaft) from the potentiometer to the control input signal (which contains information of the desired position of the shaft), and any difference between the actual and desired values (known as an error signal) is amplified and used to drive the DC motor in a direction necessary to reduce or eliminate the error. The error is zero when the output shaft gets to the desired position. The functioning block diagram of a typical servomotor is shown below [27].

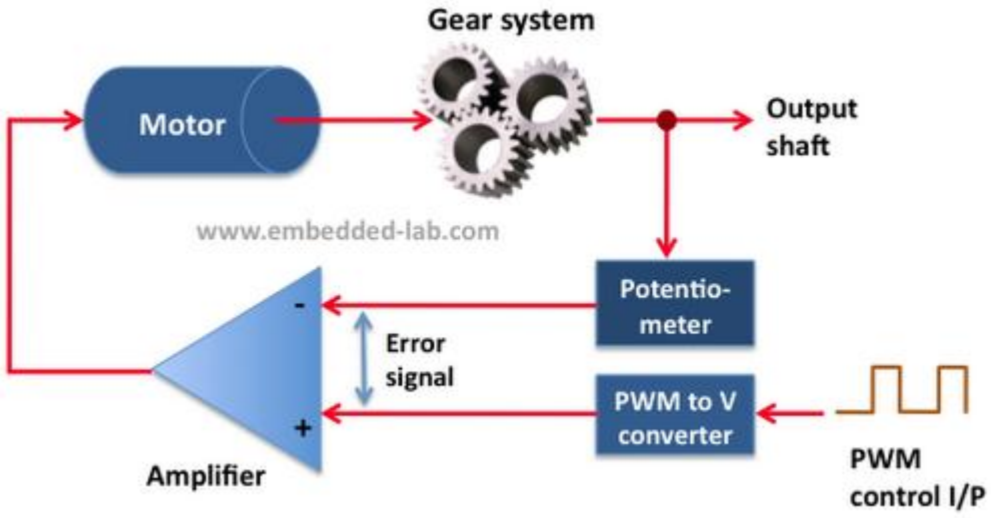


Figure 2.3 Principle of a Servo Motor [28]

The control input to a servo is a pulse width modulated (PWM) signal, generally of frequency 50 Hz. This means the pulse should repeat every 20ms. The width of the pulse determines the angular position of the output shaft. An electronic circuit inside the servo converts the PWM signal to a proportional output voltage which is compared with the feedback voltage from the potentiometer. If a difference exists between the two, the control circuit drives the motor in an appropriate direction until the difference becomes zero. A typical value of the pulse width is somewhere in the range of 1.0 to 2.0 milliseconds (ms). For a standard servo, a pulse width between 1.0 ms to 1.5 ms makes the servo to turn clockwise (CW), between 1.5 ms to 2.0 ms makes it to turn counterclockwise (CCW), and a 1.5 ms pulse width turns the servo motor to its center. However, these values could vary depending on the brand and make of the motor. It is advised to read the datasheet of the servo to find the true values of the pulse widths required for positioning the servo at different angles [28].

A servomotor has three wires: two are designated for power supply (Vcc and Ground) and the third wire is for the control signal. The Vcc wire is usually red and the ground one is either black or brown. The control signal wire comes in white, yellow, or orange color. The servomotor used in this project has red, black, and yellow color wires for Vcc, Gnd, and control signal, respectively. It operates at 5.0 V power supply and provides angular rotation through 180° [29].

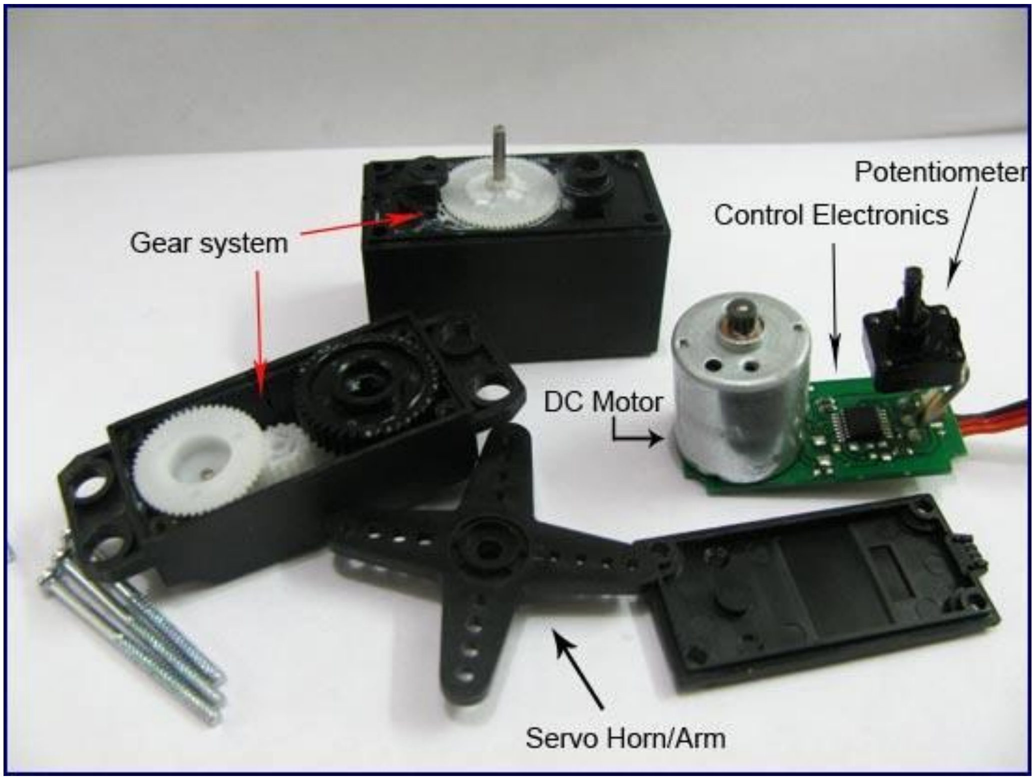
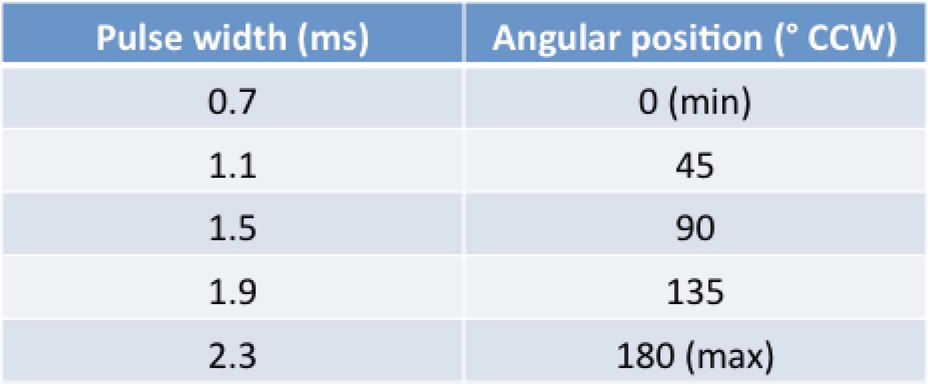


Figure 2.4 Servo Motor Construction [28]

The pulse width values for different angular positions of this servo are depicted in the table below. The repetition rate of the pulse should be 50 Hz (period of 20 ms).

Table 2.1 Servo timing information for different angular positions [28]



Pulse width (ms)	Angular position (° CCW)
0.7	0 (min)
1.1	45
1.5	90
1.9	135
2.3	180 (max)

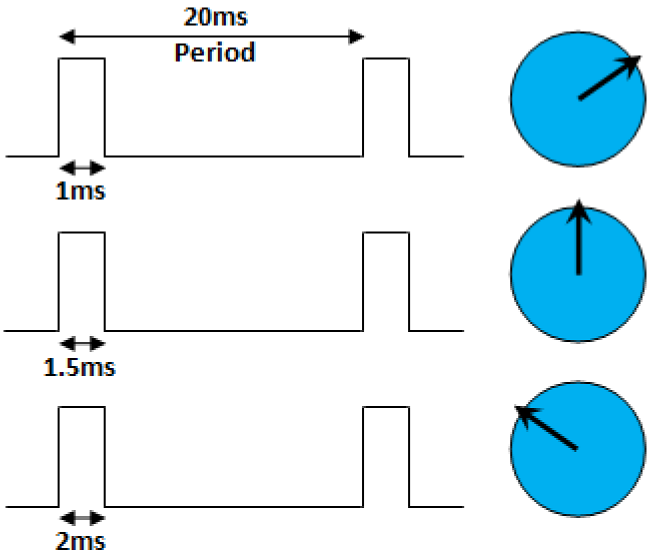


Figure 2.5 Different angular positions of the servo arm [28]

2.9 Peripheral Interface Controller (PIC) Microcontrollers

The PIC microcontroller was developed by General Instruments in 1975. PIC was developed when Microelectronics Division of General Instruments was testing its 16-bit CPU CP1600. Although the CP1600 was a good CPU but it had low I/O performance. The PIC controller was used to offload the I/O the tasks from CPU to improve the overall performance of the system.

In 1985, General Instruments converted their Microelectronics Division to Microchip Technology. PIC stands for Peripheral Interface Controller. The General Instruments used the acronyms Programmable Interface Controller and Programmable Intelligent Computer for the initial PICs (PIC1640 and PIC1650).

In 1993, Microchip Technology launched the 8-bit PIC16C84 with EEPROM which could be programmed using serial programming method. The improved version of PIC16C84 with flash memory (PIC18F84 and PIC18F84A) hit the market in 1998 [36].

2.9.1 Development of PIC

Since 1998, Microchip Technology continuously developed new high performance microcontrollers with new complex architecture and enhanced in-built peripherals. PIC microcontroller is based on Harvard architecture. At present PIC microcontrollers are widely used for industrial purpose due to its high performance ability at low power consumption. It is also very famous among hobbyists due to moderate cost and easy availability of its supporting software and hardware tools like compilers, simulators, debuggers etc [36]. The 8-bit PIC microcontroller is divided into following four categories on the basis of internal architecture:

1. Base Line PIC

   Base Line PICs are the least complex PIC microcontrollers. These microcontrollers work on 12-bit instruction architecture which means that the word size of instruction sets are of 12 bits for these controllers. These are smallest and cheapest PICs, available with 6 to 40 pin packaging. The small size and low cost of Base Line PIC replaced the traditional ICs like 555, logic gates etc. in industries [36].

2. Mid-Range PIC

   Mid-Range PICs are based on 14-bit instruction architecture and are able to work up to 20 MHz speed. These controllers are available with 8 to 64 pin packaging. These microcontrollers are available with different peripherals like ADC, PWM, Op-Amps and different communication protocols like USART, SPI, I2C (TWI), etc. which make them widely usable microcontrollers not only for industry but for hobbyists as well [36].

3. Enhanced Mid-Range PIC

   These controllers are enhanced version of Mid-Range core. This range of controllers provides additional performance, greater flash memory and high speed at very low power consumption. This range of PIC also includes multiple peripherals and supports protocols like USART, SPI, I2C and so on [36].

4. PIC18

   PIC18 range is based on 16-bit instruction architecture incorporating advanced RISC architecture which makes it highest performer among the all 8-bit PIC families. The PIC18 range is integrated with new age communication protocols like USB, CAN, LIN, Ethernet (TCP/IP protocol) to communicate with local and/or internet based networks. This range also supports the connectivity of Human Interface Devices like touch panels etc [36].

2.9.2 PIC Architecture

PIC microcontrollers are based on advanced RISC architecture. RISC stands for Reduced Instruction Set Computing. In this architecture, the instruction set of hardware gets reduced which increases the execution rate (speed) of system.

PIC microcontrollers follow Harvard architecture for internal data transfer. In Harvard architecture there are two separate memories for program and data. These two memories are accessed through different buses for data communication between memories and CPU core. This architecture improves the speed of system over Von

Neumann architecture in which program and data are fetched from the same memory using the same bus. PIC18 series controllers are based on 16-bit instruction set [31].

The question may arise that if PIC18 are called 8-bit microcontrollers, then what about them being based on 16-bit instructions set. ‘PIC18 is an 8-bit microcontroller’ this statement means that the CPU core can receive/transmit or process a maximum of 8-bit data at a time. On the other hand the statement ‘PIC18 microcontrollers are based on 16-bit instruction set’ means that the assembly instruction sets are of 16-bit [32].

The data memory is interfaced with 8-bit bus and program memory is interfaced with 16-bit bus as depicted in the following figure.

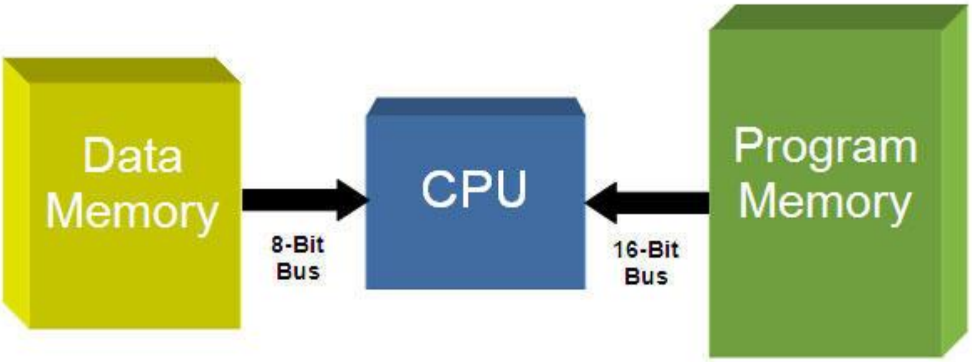


Figure 2.6 PIC18 Data Memory[32]

PIC microcontroller contains an 8-bit ALU (Arithmetic Logic Unit) and an 8-bit Working Register (Accumulator). There are different GPRs (General Purpose Registers) and SFRs (Special Function Registers) in a PIC microcontroller. The overall system performs 8-bit arithmetic and logic functions. These functions usually need one or two operands. One of the operands is stored in WREG (Accumulator) and the other one is stored in GPR/SFR. The two data is processed by ALU and stored in WREG or other registers [33].

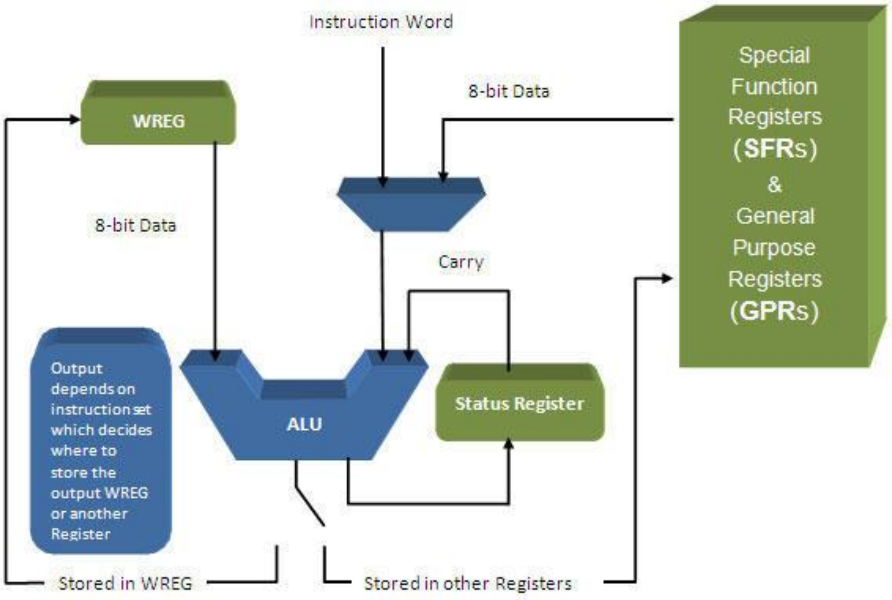


Figure 2.7 Special Function Registers [33]

The above process occurs in a single machine cycle. In PIC microcontroller, a single machine cycle consists of 4 oscillation periods. Thus an instruction needs 4 clock periods to be executed. This makes it faster than other 8051 microcontrollers [33].

Early processors and controllers could fetch or execute a single instruction in a unit of time. The PIC microcontrollers are able to fetch and execute the instructions in the same unit of time thus increasing their instruction throughput. This technique is known as instruction pipelining where the processing of instructions is split into a number of independent steps [34].

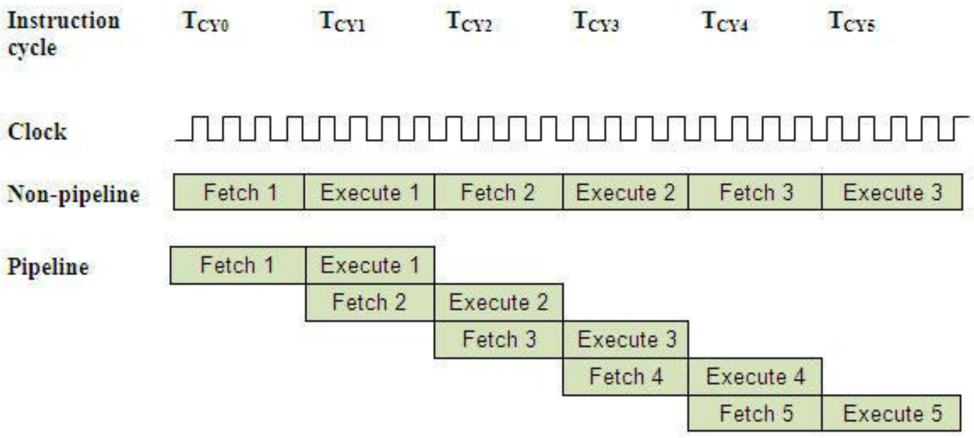


Figure 2.8 Pipelining [33]

For the purpose of this project, PIC18F4550 is chosen as the main PIC microcontroller used to demonstrate the energy management system of a parallel hybrid electric vehicle. PIC18F4550 is chosen because of the features mentioned below:

- C Compiler Optimized Architecture with Optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash
- Program Memory Typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory Typical
- Flexible oscillator option
    a. Four Crystal modes, including High-Precision PLL for USB
    b. Two External Clock modes, Up to 48 MHz
    c. Internal Oscillator: 8 user-selectable frequencies, from 31 kHz to 8 MHz
    d. Dual Oscillator Options allow Microcontroller and USB module to Run at different Clock Speeds [35]

The PIC18F4550 microcontroller also consists of following peripherals:

- I/O Ports: PIC18F4550 have 5 (PORTA, PORTB, PORTC, PORTD and PORTE) 8-bit input-output ports. PortB&PortD have 8 I/O pins each. Although other three ports are 8-bit ports but they do not have eight I/O pins. Although the 8-bit input and output are given to these ports, but the pins which do not exist, are masked internally.
- Memory: PIC18F4550 consists of three different memory sections:
    a. Flash Memory: Flash memory is used to store the program downloaded by a user on to the microcontroller. Flash memory is non-volatile, i.e., it retains the program even after the power is cut-off. PIC18F4550 has 32KB of Flash Memory.
    b. EEPROM: This is also a nonvolatile memory which is used to store data like values of certain variables. PIC18F4550 has 256 Bytes of EEPROM.
    c. SRAM: Static Random Access Memory is the volatile memory of the microcontroller, i.e., it loses its data as soon as the power is cut off. PIC18F4550 is equipped with 2 KB of internal SRAM.
- Oscillator: The PIC18F series has flexible clock options. An external clock of up to 48 MHz can be applied to this series. These controllers also consist of an internal oscillator which provides eight selectable frequency options varying from 31 KHz to 8 MHz.
- 8x8 Multiplier: The PIC18F4550 includes an 8 x 8 multiplier hardware. This hardware performs the multiplications in single machine cycle. This gives higher computational throughput and reduces operation cycle & code length.
- ADC Interface: PIC18F4550 is equipped with 13 ADC (Analog to Digital Converter) channels of 10-bits resolution. ADC reads the analog input, for example, a sensor input and converts it into digital value that can be understood by the microcontroller.

- Timers/Counters: PIC18F4550 has four timer/counters. There is one 8-bit timer and the remaining timers have option to select 8 or 16 bit mode. Timers are useful for generating precision actions, for example, creating precise time delays between two operations.
- Interrupts: PIC18F4550 consists of three external interrupts sources. There are 20 internal interrupts which are associated with different peripherals like USART, ADC, Timers, and so on.
- EUSART: Enhanced USART (Universal Synchronous and Asynchronous Serial Receiver and Transmitter) module is full-duplex asynchronous system. It can also be configured as half-duplex synchronous system. The Enhanced USART has the feature for automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These features make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.
- ICSP and ICD: PIC18F series controllers have In Circuit Serial Programming facility to program the Flash Memory which can be programmed without removing the IC from the circuit. ICD (In Circuit Debugger) allows for hardware debugging of the controller while it is in the application circuit.
- SPI: PIC18F supports 3-wire SPI communication between two devices on a common clock source. The data rate of SPI is more than that of USART.
- I2C: PIC18F supports Two Wire Interface (TWI) or I2C communication between two devices. It can work as both Master and Slave device.
- USB: PIC18F supports full-speed USB with different clock options [35].

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will describes how the project is conducted and the detailed description of the methods and steps taken in order to complete the project. The hardware and software used during the construction of this project is also discussed.

3.2 Procedure

The overall workflow for this project consists of 4 phases. The first phase is the phase includes all the researches read from journals and other researches and also project proposal. The second phase is the program development phase. This phase includes the PIC programs done in order to manage the two energy sources of a parallel hybrid electric vehicle. The third phase is the hardware development phase, whereby the PIC programs are burnt into the PICs and the circuits involved are constructed. The fourth and the last phase would the testing and running phase. This is the phase where the hardware constructed is tested to match the instructions given in the software. The energy management system of a parallel hybrid electric vehicle is also verified according to the scopes determined at an early stage of this project.

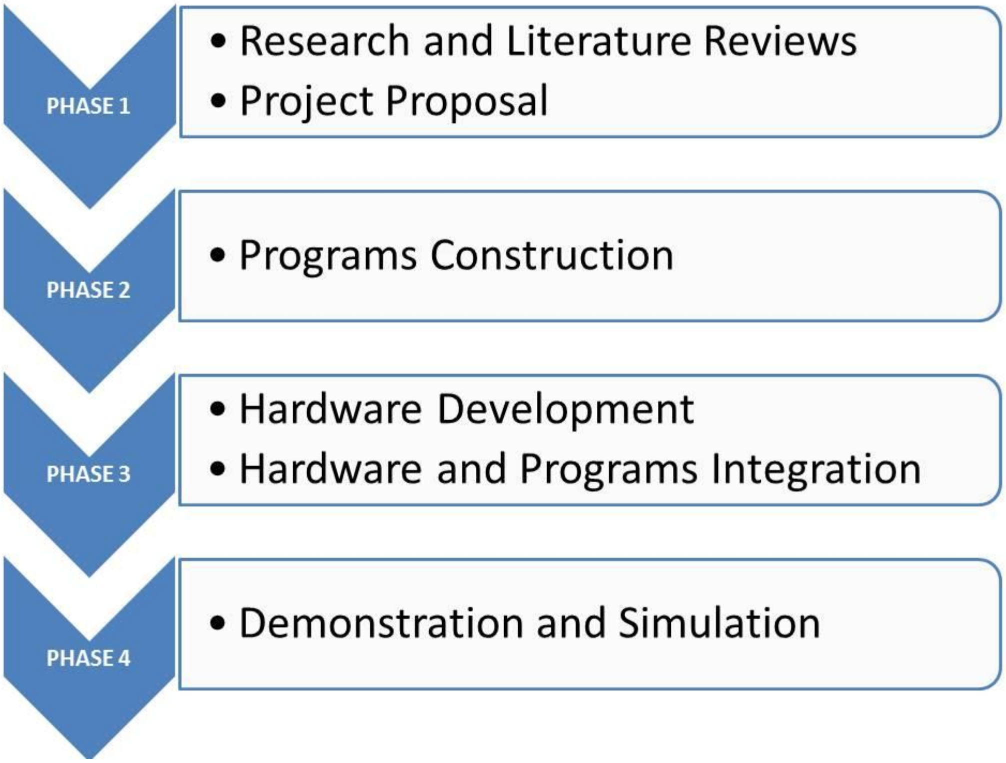


Figure 3.1 Project Development Phase

The complete process flow of this system is depicted in the flowchart in the figure below. This system is expected to operate according to the flow chart.

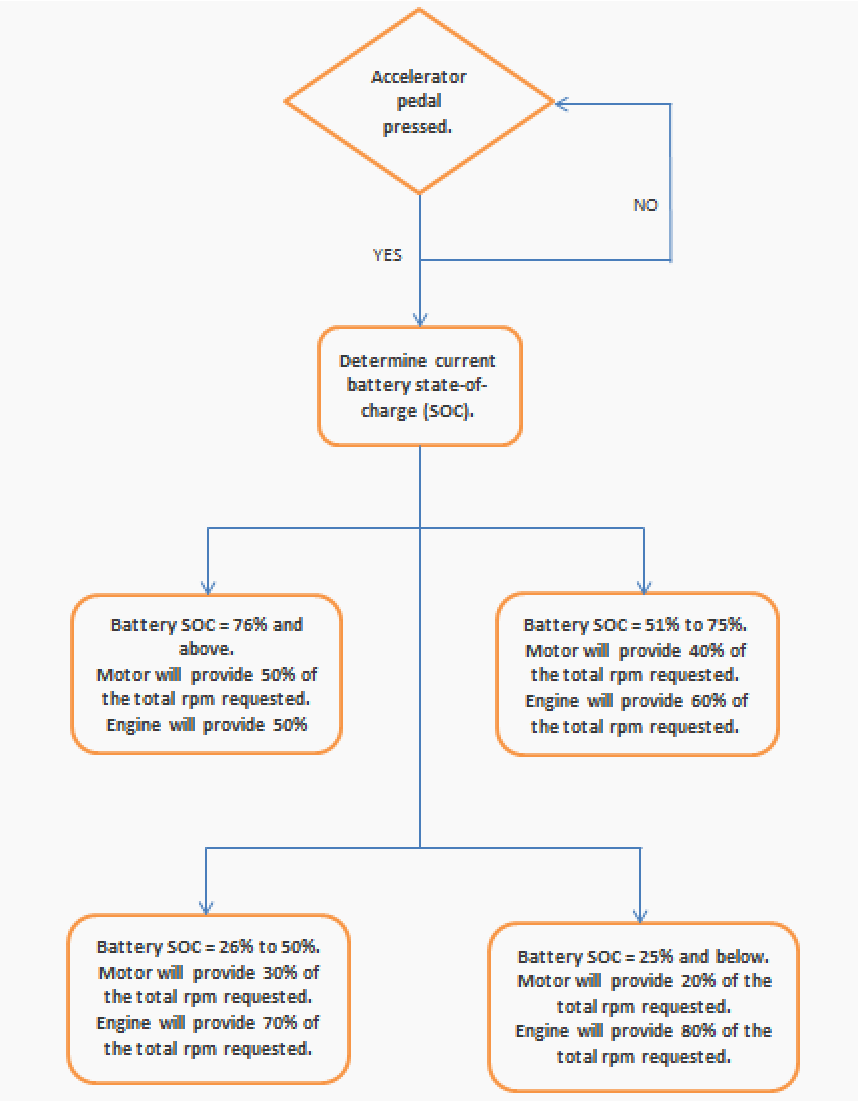


Figure 3.2 Project Process Flow

3.2.1 Programs Construction

The programs for this project are divided into two parts. One part of the programs would be the program to control the speed of the electric motor. The other part of the program is the program to control the hobby kit servo motor of a remote-controlled car.

3.2.1.1 DC Motor Speed Controller Program

The heart of the PIC18F14K50 pulse steering PWM mode is relying on the TIMER2 peripheral, where it is used as the basic counter generator for the PWM signal. The TIMER2 counter clock (TMR2) is supplied by selectable prescale clock. This prescale circuit will divide the system clock by 1, 4 or 16 respectively. The prescale could be selected by assigning the T2CKPS1 and T2CKPS0 bits in the T2CON register.

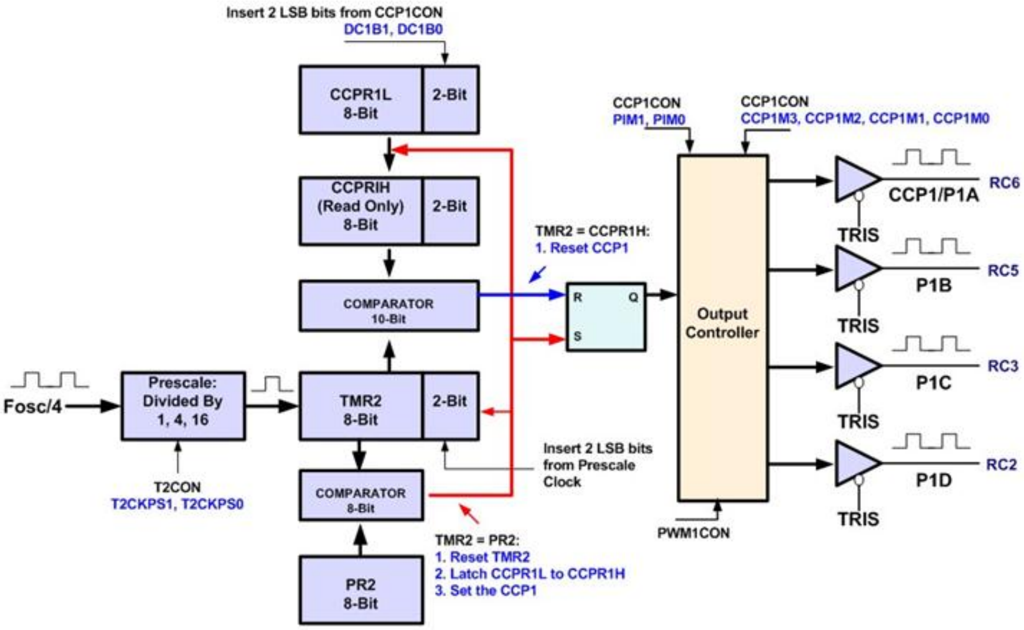


Figure 3.3 Simplified PIC18F14K50 Enhanced PWM Peripheral Circuit Diagram

The TMR2 register value is continuously compared to the PR2 register which determine the top value of the TMR2 counter register. When the TMR2 register value reaches the PR2 value, then the TMR2 counter register value will be reset back to 0.

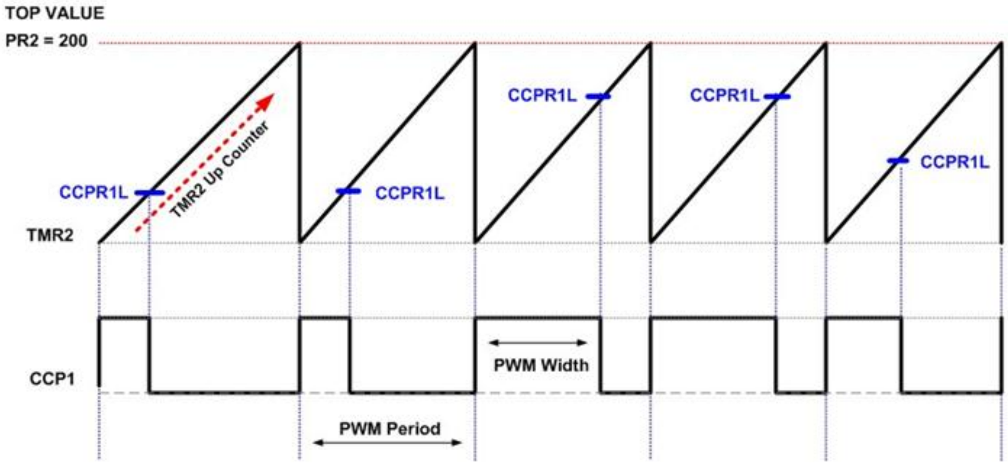


Figure 3.4 The PIC18F14K50 Enhanced PWM Output Diagram

At the same time the value of TMR2 counter register is also being compared to the CCPR1L register value. Therefore by changing the PR2 value, we could change the PWM period and this mean changing the PWM frequency as well. The PWM period could be calculated using this following formula:

PWM period = 4 x Tosc x (PR2 + 1) x (TMR2 prescale value) second

Where Tosc is the system clock period in second

PWM frequency = 1 / PWM Period Hz

By assigning the PR2 register with 200 and select the prescale to 4; and applying all these values to the formula above, the PWM frequency for the DC motor is determined based on the internal system oscillator of 16 MHz as follow:

PWM period = 4 x (1 / 16.000.000) x 201 x 4 = 0.000201 second

Therefore the PWM frequency is:

PWM frequency = 1 / 0.000201 = 4.975 kHz

The T2CON (TIMER2 Control) register is used select the postscale (T2OUTPS<3:0>), activate the TIMER2 peripheral (TMR2ON) and set the prescale clock used by the TMR2 counter register. By setting theT2CKPS1=0 and T2CKPS0=1 in the T2CON register, the 1:4 prescale is selected; and by setting theTMR2ON to logical “1”, TIMER2 peripheral is activated.

By setting P1M1=0 and P1M0=0 bits in the CCP1CON register, the single output PWM is selected; and by setting the CCP1M3=1, CCP1M2=1, CCP1M1=0 and CCP1M0=0 in the CCP1CON register, PWM mode with P1A, P1C active-high; P1B, P1D active-high is selected.

By applying the analog value read from the 10K potentiometer connected to the PIC18F14K50 RA4 pins, the PWM duty cyclecould be easily varied by changing the voltage divider output formed by the 10Kpotentiometer.

To count the DC motor rotation per minute (RPM), an infra-red reflective object sensor is used; as this type of sensor will make sensing the DC motor rotation becomes easier.

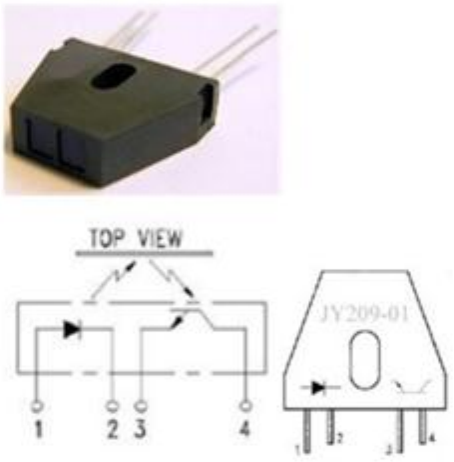


Figure 3.5 Infra-red Reflective Object Sensor

The infra-red reflective object sensor works by simply emitting the infra-red beam and when it encounters the white object surface, the infra-red beam will be reflected back to the phototransistor; next the phototransistor and the 2N3904 transistor which formed the Darlington pair will start to conduct and will generate enough voltage across the 470 Ohm resistor to be considered by the PIC18F14K50 microcontroller built in Schmitt trigger RC0 input port as the logical “1”. When the infra-red beam encounters the black tire surface than both of the phototransistor and 2N3904 transistor will turn off; and the voltage across 470 Ohm resistor will drop to zero volt (logical “0“).

Therefore by timing the generated pulse period by the infra-red reflective object sensor, the RPM could be calculated by using this following formula:

Frequency = 1/T Hz; T is the generated pulse period in second.

RPM (Rotation per Minute) = Frequency x 60

In order to calculate the DC motor RPM, the period of the pulse generated by the RPM sensor shown above is calculated. One of the methods to measure the pulse period is to use the PIC18F14K50 microcontroller ECCP (Enhanced Capture/Compare/PWM) peripheral in the capture mode to calculate the period; in the capture mode, the 16-bit TIMER1 or TIMER3 is used to count the pulse period by feeding the pulse directly to the CCP1 pins (RC5). The capture interrupt will be generated every rising edge of the pulse (or falling edge), therefore by knowing the exact TIMER1 or TIMER3 counter clock time period and get the timer 16-bit counted value between the two rising edge pulse, the RPM of the motor could be calculated.

Since the Microchip PIC18F14K50 microcontroller ECCP peripheral has been used to generate the PWM signal for the DC motor in this project, the PIC18F14K50 external interrupt peripheral on pin INT0(RC0) or INT1 (RC1) is utilized instead. This external interrupt peripheral will generate interrupt on every rising edge (or falling edge) of the pulse; therefore by combining it with the 16-bit TIMER0 counter mode, the RPM of the motor could be calculated as shown in the following diagram.

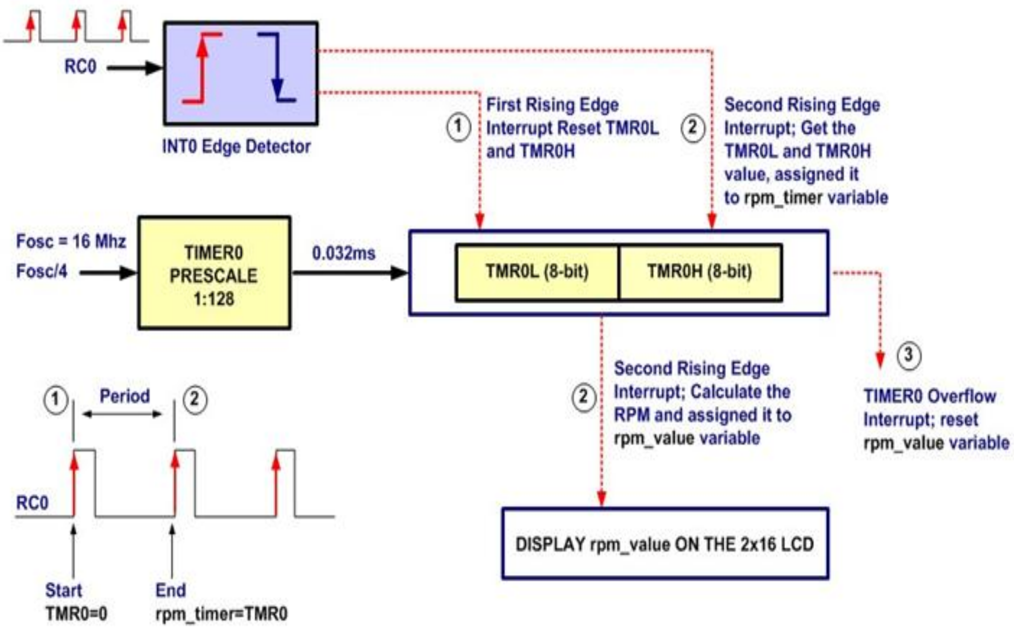


Figure 3.6 Microchip PIC18F14K50 RPM Counter Using External Interrupt (INT0) and Timer0 Diagram

As shown in the diagram above, first, the PIC18F14K50 microcontroller external interrupt has to be activated and configured to detect the pulse rising edge; the TIMER0 peripheral is configured for the RPM period counter.

By setting the TMR0E and INT0IE bits to logical "1" on the PIC18F14K50 microcontroller interrupt control register (INTCON) and TMT0ON bits to logical "1" on the TIMER0 control register (T0CON), both the TIMER0 and External peripherals are activated. The time required to increase the TIMER0 16-bit counter is calculated by selecting the 1:256 prescale value.

TIMER0 Clock period = 4 x Tosc x TMR2 prescale value second

TIMER0 Clock period = 4 x (1/16.000.000) x 128 = 0.000032 second = 0.032 ms

This means that the TIMER0 counter required 0.032 ms to increase the TMR0L and TMR0H registers counter value by one.By setting the INTEDG0 to logical “1” on INTCON2 (interrupt Control 2) register, the Rising Edge detection mode is chosen.

By resetting the TIMER0 counter on the first rising edge external interrupt and reading back the TIMER0 counter on the second rising edge of the external interrupt, the pulse period can be calculated by using the formula below;

Pulse Period = 0.032 x TIMER0 Counter (TMR0H:TMR0L) millisecond

The RPM value is the frequency of rotation measured in minute (60 second), therefore the DC motor RPM value could be calculated as the following formula:

rpm_value = (1 / Pulse Period) in second x 60 = 60000.0 / 0.032 x rpm_timer

The rpm_timer variable contains the 16-bit TIMER0 counter value, while the global rpm_value contain the RPM value of the DC motor.

The complete program for the DC motor speed controller is attached as an appendix at the end of this thesis.

3.2.1.2 Servo Motor Controller Program

The program does not require any variables to be established since the predefined B0 and B2 byte variables are used. There is nothing to initialize either. Therefore the program starts right into the main program loop. The program begins at the Center: label. This block of code centers the shaft of the servo motor. The PIC is instructed to send the PULSOUT signal 100 times by using a FOR-NEXT loop using variable B2. A 20 millisecond of pause is also applied to allow the servo motor to react.

The program PULSOUT command has a 10 microsecond resolution. The period constant that is used in the center block of the code needs to result in a 1.5 millisecond pulse. Therefore the actual value used in the command is 150(150 * 10 microsecond = 1500 microsecond, or 1.5 milliseconds).

This command is deliberately sent 100 times because it is found that by doing so, there is enough time to pull the linkage arm off of the servo and position it back in the motor shaft at the center while the motor was being driven to center.

Once the Center: loop is complete, the program moves into the servo: label block. This is where the PIC16F876 is set to drive the servo motor back and forth between the full counter-clockwise range of positions and the full clockwise range of positions. This is done with two FOR-NEXT loops. The first FOR-NEXT loop increments variable B0 by 1, starting with 100 and ending with 200. These are the end points for the servomotor. When the servo: label is approached in the program, the servo will drive immediately from the center position to most counter-clockwise position. Then the servo motor will slowly step the servo shaft to the full clockwise position. Then for the next FOR-NEXT loopincrements the servo motor in the opposite direction by changing the B0 variable from 200 to 100 in -1 steps.

After the FOR-NEXT command is completed, the GOTO statement is used to jump back up to the servo: label and the process is repeated all over again, thus creating a back and forth movement of the servomotor shaft.

The complete program is appended in the appendices.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter will describes how the hardware assembled will conform to the expected outcomes mentioned on the earlier stage of the project. The outcome expected is that the in-house model of the vehicle should divide and manage the energy sources accordingly based on the percentage of the stage of charge of the battery.

4.2 Expected Outcomes

On the earlier stage of the project development, the energy management system is developed so that these results below can be obtained:

- When the battery state-of-charge is at 100%, the energy management system will divide the workload to propel the vehicle so that 75% of the total rpm requested by the vehicle is provided by the DC motor and 25% of the total rpm requested is provided by the engine.
- When the battery state-of-charge is at 75%, the energy management system will divide the workload to propel the vehicle so that 50% of the total rpm requested by the vehicle is provided by the DC motor and 50% of the total rpm requested is provided by the engine.
- When the battery state-of-charge is at 50%, the energy management system will divide the workload to propel the vehicle so that 25% of the total rpm requested by the vehicle is provided by the DC motor and 75% of the total rpm requested is provided by the engine.
- When the battery state-of-charge is 25%, the energy management system will cut off the access the battery and this in turn will shift the workload to provide the total rpm requested by the vehicle to be only from the engine.

4.3 Results

In order to test the conformity of the system developed to the expected outcomes, the system is fed with different voltage stage to mimic the different state-of charge of the battery. Table 4.3 below summarizes the process.

Table 4.1 Test Run Results

Battery Voltage	State-of-charge (SOC)	DC Motor RPM	Engine RPM	Percentage of RPM Provided by DC Motor	Percentage of RPM Provided by Engine
5 V	100 %	0	0	0 %	0 %
3.75 V	75 %	0	0	0 %	0 %
2.5 V	50 %	0	0	0 %	0 %
1.25 V	25 %	0	0	0 %	0 %

Table 4.1 above depicts the outcomes of the project developed. In order to test the conformity of the system developed to the expected outcomes mentioned, the total requested rpm of the vehicle is fixed throughout the test. The test results obtained above depicted that the system is not functioning according to the objectives of the study set at an earlier stage of this project.

4.4 Failure Mode Analysis

The failure mode analysis is done to determine and confirm the disconformity of the test results to the expected outcomes. In order for the analysis to be carried out, a simple instruction execution test is done on the microcontrollers used.

4.4.1 Simple Instruction Execution Test

In this test, a simple digital input and output program is constructed as below:

```
void main()
{

set_tris_b(0xFF);
set_tris_c(0x00);
set_tris_d(0x00);

output_b(0xFF);
output_c(0x00);
output_d(0x00);

while (TRUE)

{
```

```
if (!input(BUTTON1))
{
output_high(LED1);
}

else if (!input(BUTTON2))
{
output_high(LED2);
}

else if (!input(BUTTON3))
{
output_high(LED3);
}

else
{
}
}
}
```

The program constructed above will determine the reliability of the PICs used in executing instructions. The results obtained from the test are depicted in the table below:

Table 4.2 Program Execution Test Result

PIC	Button pressed	LED lit	Output voltage
1	Button 1	None	0.42 V
2	Button 2	None	0.53 V
3	Button 3	None	0.46 V

Based on the result above, it is clearly depicted that all three PICs used are not executing the program downloaded. It is observed that when any of the buttons are pushed, no LED is lit. The output voltage obtained at each and every output pin does not conform to the expected amount of voltage, which is 5V.

Therefore, in the matter of the fact of the results depicted above, it is assumed that the three PIC microcontrollers used are faulty, hence contributing to the disconformity of the test results with the expected outcomes.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This project aims to develop an energy management system of a parallel hybrid electric vehicle. The system is developed based on a small scale model of an in-house parallel hybrid electric vehicle. Based on the fact that hybrid vehicles are significant in today’s world where sustainable inventions are desirable, this project proves to be an important addition in learning more about the principle of how energy sources are managed in a hybrid electric vehicle.

The expected results of this project are not achieved according to the scopes set at the early stage of the project development. The control circuit developed does not divides the workload as per requested by the vehicle accordingly between the two energy sources of the vehicle model, based on the SOC of the battery.

5.2 Recommendations for Future Development

Although this project does not achieve its targeted objectives, there are still numerous additions and developments that could be added in order to develop a complete fully functioning in-house parallel hybrid electric vehicle. The suggestions and recommendations for future extension of this project are as below:

- Since this model of hybrid vehicle does not actually moves in the sense that the mechanical parts of the vehicle are left out so that the moving parts of the model are only the DC motor and the engine, future development of this model could focuses on transforming this model into a model that would actually moves on all tires.
- This model of a parallel hybrid electric vehicle does not include a Graphical-User-Interface or GUI to depicts the process of which is happening during the energy division. In order to get a clearer view on the division system, a GUI system could be developed in the future.
- It is also suggested the PIC microcontrollers used are to be handled with care and not to be harshly operated. This is because PIC microcontrollers are sensitive to over exposure to ultraviolet light and they can also be damaged if they are subjected to electrostatic charge [37].

5.3 Cost and Potential Commercialization

The overall cost of implementation of this project is as depicted in the table below:

Table 5.1 Materials Cost

Materials	**Cost**
Radio-controlled Car	RM500
Electronics components	RM186.90

The total cost of implementing this study is RM686.90. Based on the failure mode analysis done, it is favorable that this project be commercialized as assistance in studying about parallel hybrid electric vehicle. Even though the project is not functioning according to the scopes set, the glitch in the system can be further analyzed and solved for future study.

REFERENCES

1. http://wikipedia.org/wiki/Ferdinand _Porsche

2. http://www.hybridcars.com/history/history-of-hybrid-vehicles.html

3. http://en.wikipedia.org/wiki/Voiturette

4. Moore, T.C. (1996), “Tools and Strategies for Hybrid-electric Drive System Optimization,” SAE Technical Paper Series, No. SP-1189.

5. Fellini, R., Michelena, N., Papalambros, P. et al. (1999), “ Optimal design of automotive hybrid powertrain systems, in Proceedings of EcoDesign 99 – First International Symposium on Environmentally Conscious Design and Inverse Manufacturing,” (ed. H. Yoshikawa et al.), Tokyo, pp. 400–405.

6. Gao, W. and Porandla, S. (2005), “ Design optimization of a parallel hybrid electric powertrain,” Proceedings of the IEEE Vehicle Power and Propulsion Conference, Chicago, September, pp. 530–535.

7. F. R. Salmasi, “Designing control strategies for hybrid electric vehicles,” in Proc. Tutorial Presentation EuroPes, Benalmadena, Spain, Jun. 15–17, 2005.

8. C. G. Hochgraph, M. J. Ryan, and H. L. Wiegman, “Engine control strategy for a series hybrid electric vehicle incorporating load leveling and computer controlled energy management,” SAE J. SAE/SP-96/1156, pp. 11–24, 2000.

9. B. M. Baumann, G. Washington, B. C. Glenn, and G. Rizzoni, “Mechatronic design and control of hybrid electric vehicles,” IEEE/ASME Trans. Mechatronics, vol. 5, no. 1, pp. 58–71, Mar. 2000.

10. H.-D. Lee and S.-K.Sul, “Fuzzy-logic-based torque control strategy for parallel-type hybrid electric vehicle,” IEEE Trans. Ind. Electron., vol. 45, no. 4, pp. 625–632, Aug. 1998.

11. C. Mi, M. A. Masrur and D. W. Gao, “Hybrid Electric Vehicles: Principles and Applications with Practical Perspectives,” Wiley.

12. N. J. Schouten, M. A. Salman and N. A. Kheir.“Energy management strategies for parallel hybrid vehicles using fuzzy logic,” Department of Electrical and Systems Engineering, Oakland University and General Motors Research Development Center.

13. http://www.ermicro.com/blog/?p=1461

14. I. Dogan, “PICBASIC Projects - 30 Projects Using PICBASIC and PICBASIC PRO”

15. Mohd R. Mohamed, Suleiman M. Sharkh and Frank C. Walsh, “Redox Flow Batteries for Hybrid Electric Vehicles: Progress and Challenges” School of Engineering Sciences, University of Southampton andFaculty of Electrical & Electronics Engineering, Universiti Malaysia Pahang.

16. N. Armaroli and V.Balzani, “Energy for a Sustainable World: From the Oil Age to a Sun-Powered Future,” WILEY-VCH GmbH & Co. KGaA.

17. “History of the electric vehicle,” Available: http://en.wikipedia.org/wiki/History_of_the_electric_vehicle, Retrieved on June 02, 2012, Accessed on June 02, 2012.

18. D. J. Santini, “Electric Vehicle Waves of History: Lessons Learned about Market Deployment of Electric Vehicles,” Argonne National Laboratory.

19. J.Gover, “A Tutorial on Hybrid Electric Vehicles: EV, HEV, PHEV and FCEV,” Electrical Engineering Department, Kettering University.

20. R. Q. Riley, “Electric and Hybrid Vehicles: An Overview of the Benefits, Challenges, and Technologies,” Available: http://www.rqriley.com/ev-tech.htm, Retrieved on June 02, 2012, Accessed on June 02, 2012.

21. M.Ehsani, YiminGao and and John M. Miller, “Hybrid Electric Vehicles: Architecture and Motor Drives.”

22. “Radio-controlled car,” Available: http://en.wikipedia.org/wiki/Radio-controlled_car, Retrieved on June 07, 2012, Accessed on June 07, 2012.

23. “What are Radio Control (RC) Cars?,” Available: http://www.hpieurope.com/faq.php?lang=en&catId=whatis, Retrieved on June 07, 2012, Accessed on June 07, 2012.

24. “How Radio Controlled Toys Work?,” Available: http://electronics.howstuffworks.com/rc-toy.htm, Retrieved on June 08, 2012, Accessed on June 08, 2012.

25. Meng Lee, “Operation principles for an RC car,” Available: http://www.articlesfactory.com/articles/entertainment/operation-principles-for-an-rc-car.html, Retrieved on June 08, 2012, Accessed on June 08, 2012.

26. “Nitro engine,” Available: http://en.wikipedia.org/wiki/Nitro_engine, Retrieved on June 15, 2012, Accessed on June 15, 2012.

27. “Servo motor,” Available: http://en.wikipedia.org/wiki/Servo_motor, Retrieved on June 15, 2015, Accessed on June 15, 2012.

28. “Servomechanism,” Available: http://en.wikipedia.org/wiki/Servomechanism, Retrieved on June 16, 2012, Accessed on June 16, 2012.

29. “What’s a servo,” Available: http://www.seattlerobotics.org/guide/servos.html, Retrieved on June 16, 2012, Accessed on June 16, 2012.

30. “Servo Motor,” Available: http://www.beam-wiki.org/wiki/Servo_Motor, Retrieved on June 16, 2012, Accessed on June 16, 2012.

31. “PIC Microcontroller,” Available: http://www.engineersgarage.com/articles/pic-microcontroller-tutorial, Retrieved on June 17, 2012, Accessed on June 17, 2012.

32. “PIC microcontroller”, Available: http://en.wikipedia.org/wiki/PIC_microcontroller, Retrieved on June 17, 2012, Accessed on June 17, 2012.

33. A. R. Tamuri and Y. M. Daud, “Introduction to PIC Microcontroller,” Physics Department, Faculty of Science, UniversitiTeknologi Malaysia.

34. “Peripheral Interface Controller (PIC),” Available: http://thbelectronics.blogspot.com/2011/12/peripheral-interface-controllerpic.html, Retrieved on June 17, 2012, Accessed on June 17, 2012.

35. “Introduction to Peripheral Interface Controllers (PIC),” Available: http://www.circuitstoday.com/peripheral-interface-controller-pic, Retrieved on June 17, 2012, Accessed on June 17, 2012.

36. “Scaling the PIC MCU &dsPIC DSC Families,” Available: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2551, Retrieved on June 17,2012, Accessed on June 17, 2012.

37. “The PIC's Poor Life Expectancy,” Available: http://www.twyman.org.uk/FAQ/PIC-Life.htm, Retrieved on June 21, 2012, Accessed on June 21, 2012.

APPENDIX A

DC MOTOR SPEED CONTROLLER PROGRAM

```
#include <pic18.h>
/*
** PIC18F14K50 Configuration Bit:
**
** FCMDIS     - Fail-Safe Clock Monitor disabled
** CPUDIV_0   - No CPU System Clock divide
** RCIO       - Internal RC Oscillator
** PLLDIS     - PLL is under software control
** ----------------------------------------------------------------------
** BORDIS     - Brown-out Reset disabled in hardware and software
** WDTDIS     - WDT is controlled by SWDTEN bit of the WDTCON register
** ----------------------------------------------------------------------
** MCLREN     - MCLR pin enabled, RE3 input pin disabled
** ----------------------------------------------------------------------
** XINSTDIS   - Disable extended instruction set (Legacy mode)
** LVPDIS     - Single-Supply ICSP disabled
*/
__CONFIG(1, FCMDIS & CPUDIV_0 & RCIO & PLLDIS);
__CONFIG(2, BORDIS & WDTDIS);
__CONFIG(3, MCLREN);
__CONFIG(4, XINSTDIS & LVPDIS);
__CONFIG(5, 0xFFFF);
__CONFIG(6, 0xFFFF);
__CONFIG(7, 0xFFFF);
// LCD Definition
#define LCD_HOME 0x02
#define LCD_NEXT_LINE 0xC0
#define LCD_CLEAR 0x01
#define LCD_1CYCLE 0
#define LCD_2CYCLE 1
// RPM Counter Variable
volatile unsigned intrpm_value;
charsdigit[6]={'0','0','0','0','0','\0'};
/* Delay Function */
#define FOSC 16000000UL  // Using Internal Clock of 16 MHz
#define  delay_us(x) { unsigned char _dcnt; \
                         _dcnt = (x)/(24000000UL/FOSC)|1; \
                  while(--_dcnt != 0) continue; \
                    }
```

```
voiddelay_ms(unsigned intcnt)
{
unsigned char i;
do {
i = 5;
do {
delay_us(164);
    } while(--i);
  } while(--cnt);
}
// PIC18 High-priority Interrupt Service
void interrupt high_isr(void){
static unsigned char pulse_state=0;
unsignedintrpm_timer;
if (TMR0IF) {                     // Check for TIMER0 Overflow Interrupt
rpm_value = 0;                  // Reset the RPM Value
    TMR0IF=0;                       // Clear TIMER0 interrupt flag
  }
if (INT0IF){                      // Check for External INT0 Interrupt
switch(pulse_state) {
case 0:                       // First Low to High Pulse
        TMR0H = 0;                  // Zero the high byte in TMR0H Buffer
        TMR0L = 0;                  // Clear 16-bit TIMER0 Counter
pulse_state=1;
break;
case 1:                       // Second Low to High Pulse
rpm_timer=TMR0L;            // Get the first 8-bit TIMER0 Counter
rpm_timer+=(TMR0H << 8);    // Get the last 8-bit TIMER0 Counter
        // Calculate RPM = 60 x (1/Period)
        // RPM Value = 60000 (1 / (0.032 ms x rpm_timer))
rpm_value = (int) (60000.0 / (0.032 * rpm_timer));
pulse_state=0;
    }
    INT0IF = 0;                     // Clear INT0 interrupt flag
  }
}
/*
** LCD Routine
** LCD Data RB7,RB6,RB5,RB4
** LCD Control: RC7 -> E-Enable, RC6 -> RS-Register Select, R/W-Always 0
*/
voidLCD_putcmd(unsigned char data,unsigned char cmdtype)
{
  // Put the Upper 4 bits data
  PORTB = data & 0xF0;
```

```
  RC6=0;         // RS = 0
  RC7=1;         // E = 1
  // E=0; write data
  RC7=0;
delay_us(1);    // Delay 1us for 16 MHz Internal Clock

  // cmdtype = 0; One cycle write, cmdtype = 1; Two cycle writes
if (cmdtype) {
    // Put the Lower 4 bits data
    PORTB = (data & 0x0F) << 4;
    RC6=0;       // RS = 0
    RC7=1;       // E = 1

    // E=0; write data
    RC7=0;
delay_us(1); // Delay 1us for 16 MHz Internal Clock
  }
delay_ms(5);             // Wait for busy flag (BF)
}
voidLCD_putch(unsigned char data)
{
  // Put the Upper 4 bits data
  PORTB = data & 0xF0;
  RC6=1;         // RS = 1
  RC7=1;         // E = 1
  // E=0; write data
  RC7=0;
delay_us(1);   // Delay 1us for 16 MHz Internal Clock

  // Put the Lower 4 bits data
  PORTB = (data & 0x0F) << 4;
  RC6=1;         // RS = 1
  RC7=1;         // E = 1

  // E=0; write data
  RC7=0;
delay_ms(5);             // Wait for busy flag (BF)
}
voidLCD_init(void)
{
  // Wait for more than 15 ms after VCC rises to 4.5 V
delay_ms(30);
  // Send Command 0x30
LCD_putcmd(0x30,LCD_1CYCLE);
  // Wait for more than 4.1 ms
```

```
delay_ms(8);
  // Send Command 0x30
LCD_putcmd(0x30,LCD_1CYCLE);
  // Wait for more than 100 us
delay_us(200);          // Delay 250us for 16 MHz Internal Clock  ;
  // Send Command 0x30
LCD_putcmd(0x30,LCD_1CYCLE);
  // Function set: Set interface to be 4 bits long (only 1 cycle write).
LCD_putcmd(0x20,LCD_1CYCLE);
  // Function set: DL=0;Interface is 4 bits, N=1; 2 Lines, F=0; 5x8 dots font)
LCD_putcmd(0x28,LCD_2CYCLE);
  // Display Off: D=0; Display off, C=0; Cursor Off, B=0; Blinking Off
LCD_putcmd(0x08,LCD_2CYCLE);
  // Display Clear
LCD_putcmd(0x01,LCD_2CYCLE);
  // Entry Mode Set: I/D=1; Increament, S=0; No shift
LCD_putcmd(0x06,LCD_2CYCLE);
  // Display On, Cursor Off
LCD_putcmd(0x0C,LCD_2CYCLE);
}
voidLCD_puts(const char *s)
{
while(*s != 0) {      // While not Null
if (*s == '\n')
LCD_putcmd(LCD_NEXT_LINE,LCD_2CYCLE);  // Goto Second Line
else
LCD_putch(*s);
s++;
  }
}
// Implementing integer value from 0 to 65530
char *num2str(unsigned intnumber,unsigned char start_digit)
{
unsigned char digit;
if (number > 65530) number = 0;

digit = '0';                       // Start with ASCII '0'
while(number >= 10000)             // Keep Looping for larger than 10000
   {
digit++;                         // Increase ASCII character
number -= 10000;                 // Subtract number with 10000
   }

sdigit[0]='0';                     // Default first Digit to '0'
if (digit != '0') sdigit[0]=digit; // Put the first digit
```

```
digit = '0';                       // Start with ASCII '0'
while(number >= 1000)              // Keep Looping for larger than 1000
   {
digit++;                         // Increase ASCII character
number -= 1000;                  // Subtract number with 1000
   }
sdigit[1]='0';                     // Default Second Digit to '0'
if (digit != '0') sdigit[1]=digit; // Put the Second digit
digit = '0';                       // Start with ASCII '0'
while(number >= 100)               // Keep Looping for larger than 100
   {
digit++;                         // Increase ASCII character
number -= 100;                   // Subtract number with 100
   }
sdigit[2]='0';                     // Default Second Digit to '0'
if (digit != '0') sdigit[2]=digit; // Put the Second digit
digit = '0';                       // Start with ASCII '0'
while(number >= 10)                // Keep Looping for larger than 10
   {
digit++;                         // Increase ASCII character
number -= 10;                    // Subtract number with 10
   }
sdigit[3]='0';                     // Default Second Digit to '0'
if (digit != '0') sdigit[3]=digit; // Put the Second digit
sdigit[4]='0' + number;
return(sdigit + start_digit);
}
void main(void)
{
unsigned char motor_stat,duty_cycle;
  OSCCON=0x70;         /* Select 16 MHz internal clock */
  // Initial PORT
  TRISA = 0x30;        // Input for RA4 and RA5
  TRISC = 0x01;        // Set RC0 as Input, RC<7:1> on PORTC as Output
  PORTC = 0x00;        // Initial Port C
  TRISB = 0x00;        // Set PORTB as Output
  PORTB = 0x00;        // Initial Port B
  TRISB = 0x00;          // Set All on PORTB as Output
  ANSEL = 0x08;        // Set PORT AN3 to analog input
  ANSELH = 0x00;       // Set PORT AN8 to AN11 as Digital I/O
  // Initial LCD using 4 bits data interface
LCD_init();
LCD_puts("PICJazz 20-PIN\n");
  // Init ADC
  ADCON0=0b00001101;   // ADC port channel 3 (AN3), Enable ADC
```

```
  ADCON1=0b00000000;   // Use Internal Voltage Reference (Vdd and Vss)
  ADCON2=0b00101011;   // Left justify result, 12 TAD, Select the FRC for 16 MHz

  // Init TIMER0: Period: 4 x Tosc x Prescale for each counter
  // Tosc = 1/16 Mhz = 0.0000000625
  // TIMER0 Period: 4 x 0.0000000625 x 128 = 0.000032 Second = 0.032 ms
  T0CON = 0b10000110;   // TIMER0 Enable, use 16-bit timer and prescale 1:128
  TMR0H = 0;            // Zero the high byte in TMR0H Buffer
  TMR0L = 0;            // Clear 16-bit TIMER0 Counter
  TMR0IE = 1;           // Enable TIMER0 Overflow Interrupt
  // Set the External Interrupt on INT0 (RC0) Port
  INT0IE = 1;          // Enables the INT0 external interrupt
  INTEDG0 = 1;         // Interrupt on rising edge

  // Init PWM for Single Output
  CCP1CON=0b00001100;  // Single PWM mode; P1A, P1C active-high; P1B, P1D active-
high
  CCPR1L=0;            // Start with zero Duty Cycle
  PSTRCON=0b00000100;  // Enable PIC Pulse Steering PWM on RC3 Port
  // PWM Period = 4 x Tosc x (PR2 + 1) x TMR2 Prescale Value
  // Tosc = 1/16 Mhz = 0.0000000625
  // PWM Period = 4 x 0.0000000625 x 201 x 4 = 0.000201
  // PWM Frequency = 1/PWM Period = 1/0.000201 = 4.975 kHz
  T2CON=0b00000101;    // Postscale: 1:1, Timer2=On, Prescale = 1:4
  PR2=200;             // Frequency: 4.975 kHz
  TMR2=0;              // Start with zero Counter

  // Initial Variable used
rpm_value=0;
motor_stat=0;        // Motor Off Condition
duty_cycle=0;        // 0 Duty Cycle

  // Now Enable the Interrupt
  IPEN = 1;           // Enable High Priority Interrupt
  GIEH = 1;           // Global Interrupt Enable (High Priority)
for(;;) {
if (RA5 == 0) {   // Read Switch
delay_ms(1);
if (RA5 == 0) { // Read again for Simple Debounce
motor_stat ^= 0x01;
      }
    }

if (motor_stat) {
      GODONE=1;
```

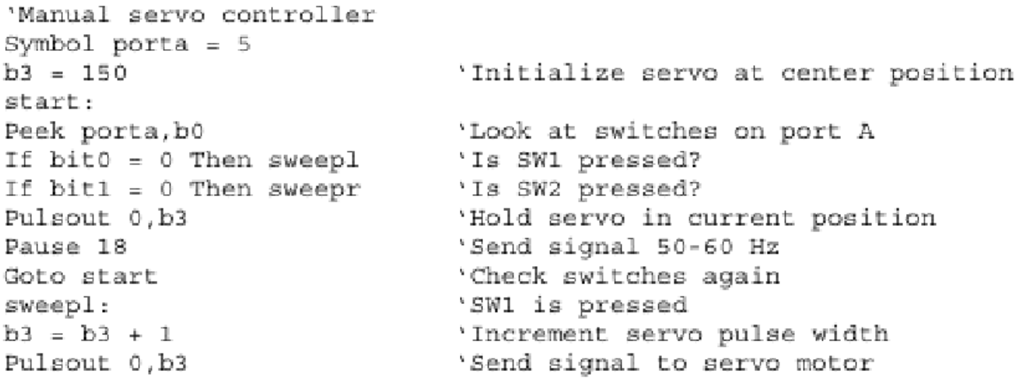
```
while (GODONE) continue; // Wait conversion done
duty_cycle=ADRESH;       // Get the High byte ADC 8-bit result
    } else {
duty_cycle=0;
    }

    // Assign duty cycle to the PWM CCPR1L register
    CCPR1L = duty_cycle;
    // Display the Information on the LCD
LCD_putcmd(LCD_HOME,LCD_2CYCLE);      // LCD Home
LCD_puts("Duty Cycle: "); LCD_puts(num2str((int)((duty_cycle/255.0) * 100.0),3));
LCD_puts(" %");
LCD_putcmd(LCD_NEXT_LINE,LCD_2CYCLE); // Goto Second Line
LCD_puts("RPM: "); LCD_puts(num2str(rpm_value,1));

    // Put the delay here
delay_ms(10);
  }
}
/* EOF: pwmrpm.c */
```

APPENDIX B

SERVO MOTOR CONTROLLER PROGRAM



```
‘Manual servo controller
Symbol porta = 5
b3 = 150                      ‘Initialize servo at center position
start:
Peek porta,b0                 ‘Look at switches on port A
If bit0 = 0 Then sweepl       ‘Is SW1 pressed?
If bit1 = 0 Then sweepr       ‘Is SW2 pressed?
Pulsout 0,b3                  ‘Hold servo in current position
Pause 18                      ‘Send signal 50-60 Hz
Goto start                    ‘Check switches again
sweepl:                       ‘SW1 is pressed
b3 = b3 + 1                   ‘Increment servo pulse width
Pulsout 0,b3                  ‘Send signal to servo motor
```