DESIGN AND DEVELOPMENT OF ELECTRIC POWERED WHEELCHAIR MODEL FOR NAVIGATION CONTROL

AFIQAH BINTI YAACOB

FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY TEKNOLOGI MARA
MALAYSIA

MAY 2010

"I declare that I have read this work and in
my opinion this work is adequate in term of scope and
quality for the purpose of awarding a Bachelor's Degree of
Electrical Engineering (Electronic) "

Signature:________________________

Supervisor's Name: Cik Rafidah Rosman

Date Approve :____________________

DECLARATION

I hereby declare that this project is my own except for quotation and summaries which have been acknowledge

ACKNOWLEDGEMENTS

First and foremost, I thank Allah S.W.T for His Blessing to let me finish this thesis. I would like to thank my Final Year Project Supervisor, Miss Rafidah Rosman for her guidance and encouragement during my study process. Special thanks to Encik Idris for his teaching and helping in develop the wheelchair model design.

Also thanks to my colleagues who always support, encourage and provided me all the information required for my project. And finally, thanks to my family, especially my parents and my sister who offered moral support, calm me and endured this long process with me.

ABSTRACT

This paper describes the significant design to build a wheelchair navigation based on joystick-controlled. This project is designed for particular purpose to increase the ease of mobility for disabled or golden ager especially those who hardly able to use both hand and legs for moving. The design would allow these people to not rely on others to move the wheelchair.

A model of electrical powered wheelchair (EPW) moves according to the motion mode of a joystick operated by one person. The wheelchair model move when user moves the joystick according to any direction. The signal from joystick is converted to digital signal by Analogue to Digital Converter (ADC). This digital signal will enable the microcontroller to perform simple control algorithm to drive the DC motor to move forward or backward. During the movement, navigation system will be interrupted by the sonar range sensor. When the sensor detect hole on the pathway, the navigation system automatically avoid the hole by commanding backward operation. The user also will be alerted about the hole by the sound made by the buzzer. In addition to all the foregoing, the design of the system was compact and small thus suitable to be implemented into real wheelchair.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURE

LIST OF ABBREVIATION

ADC	-	Analog Digital Converter
CPU	-	Central Processing Unit
DC	-	Direct Current
DIP	-	Dual In-line Package
EPW	-	Electric Powered Wheelchair
I/O	-	Input output
IC	-	Integrated Circuit
Kg	-	Kilogram
LCD	-	Liquid Crystal Display
MCLR	-	Master Clear
MCU	-	Microcontroller unit
PP	-	Peak to Peak
PWM	-	Pulse Width Modulation
RX	-	Receiver

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

The ‘Elderly’ are defined as people aged 60 years old and above. Based on population census on the year 2000, the Malaysian citizen who aged more than 65 years old was figured out to be 1.23 million people. At the end of June 2009, the Department of Social Welfare Malaysia (DSW) has reported that totals of 261,154 disabled people have registered with them. However, this figure is not a true reflection of the number of disabled people as the registration is voluntary. By taking all this value, this shows lots of people have moving impairment. From medical dictionary, impairment is described as weakening, damage, or deterioration, especially as a result of injury or disease. These people, suffering from motor deficits, disorientation, amnesia, or cognitive deficits, are dependent upon others to push them, so often feel powerless and facing difficulties in movement. Therefore, they must be given the same opportunities as everyone else to live life as independently as possible. With the aid of electric powered wheelchair such dependence is possible because the wheelchair facilitate the process of transporting the user.

1.2 Objective of Project

The main goal intended to be attained of this project is to build navigation system based on joystick. It also to show navigation based on three wheeled design wheelchair. In addition, the navigation is aided by the sonar range sensor to detect holes along the path.

1.3 Scope of Project

In order to achieve the objectives, there are several scope had been outlined. The scope of this project includes using MPLAB IDE to program microcontroller PIC 16F877A. Implementation of suitable hardware for the navigation system purpose is done, such as joystick for movement indication, motor for tires movement and sonar range sensor is used to detect holes for castor easy to move.

1.4 Outline of Thesis

This thesis consists five chapters. In first chapter, it discuss about the background, objective and scope of this project. Then, Chapter 2 tells more on theory and literature reviews that have been done. Here, discussion are about how the wheelchair move, navigation mode to be implement such as using joystick and the software to be used. In Chapter 3, the discussion will be on the methodology hardware and software implementation of this project. The result and discussion will be presented in Chapter 4. Last but not least, Chapter 5 discusses the conclusion of this project and future work that can be done.

CHAPTER 2

THEORY AND LITERATURE REVIEW

2.1 INTRODUCTION

By using the embedded computing architecture, the proposed robotic wheelchair controller performs characteristics of compact size, better reliability and lower power consumptions [13]. This chapter includes the study on the wheelchair movement based on two tires, hardware implementation (DC motor, joystick and sensor) and software implementation (MPLAB IDE and Proteus VSM 7)

2.2 WHEELCHAIR MOVEMENT

There will be two motors to be used and each motor will control each tire. The turning operation is established by:

- If turn right, right-motor stop and left-motor move forward
- If turn left, left-motor stop and right-motor move forward

The forward command indicate that both the motor move in front. Otherwise, for backward operation the motor move backward.

2.3 DC MOTOR

Basically there are three types of motor used in electrical wheelchair. Each motor is suitable for specific purpose which means some motors are better for light weight operations while others are better for heavy duty-operations.

First type is a two-pole motor which can be founded in lightweight electric powered wheelchair. It is installed for indoor use. The electricity will enters the wheelchair motor via two separate points. The chairs using this motor are capable of transporting an individual weighing up to 250 pounds or 113.5 kg.

Second type is a four-pole motor which is used in heavy-duty electric chairs. The wheelchair is therefore suitable for indoor and outdoor use. Since the electricity from battery enters the motor via four points, this result in the more power motor. Lastly, the brushless motors which have a limit number of parts. The fewer part means that fewer mechanisms make any contact when the chair is in operation. The last type of this motor is the newest electric chair motor offered and is as powerful as the four-pole motors currently found in most electric wheelchairs.

Therefore, the chosen motor is DC brushless motor is because of its enough torque to move tires [2]. As shown in Figure 2.0 below, the simple DC motor has a rectangular conductor set in a magnetic field. The field may be provided by permanent magnets in small motor, or if using a larger motor, the field is provided by winding. As a current is passed through the conductor, this will cause a cylindrical magnetic field occur around it. Then the interaction with planar magnetic field provided by the field magnet (small motor) or winding (large motor) will cause a tangential force on the rotor, which provides the motor torque.

In order for rotation to occur, the current is supplied to the armature through slip rings and brushes. To maintain the torque in the same direction, the current has to be reversed every half revolution, so the slip ring is split to form a commutator.

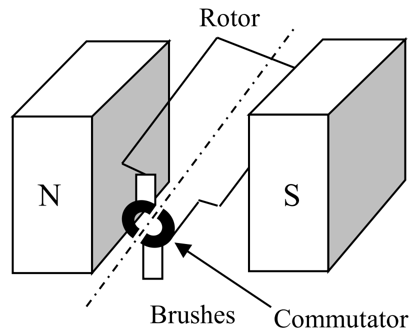

Figure 2.1 Basic motor configuration

Small DC motors typically have a small number of armatures winding with a permanent field magnet. Typically, small DC motor has three armature windings and a six-segment commutator.

2.4 JOYSTICK

There are many types of control used in the electric wheelchair. Some of the controllers are by a handle, a joystick, a motion sensitive tube, a head-to-chin controller or an eye-to-computer screen controller. The entire controllers have been used depending upon the degree of disability that the individual using the electric powered chair has. The controller is a computer interface which is responsible for amperage, speed control and the maintenance of straight line propelling and turning control when the chair is in use.

These project proposed controller based on joystick. Joystick produces analog output when the operation (forward, backward, right and left) is commanded. Thus, ADC is needed so that microcontroller can read and analysis the output from joystick.

2.5 ULTRASONIC SENSOR

In general, an autonomous robotic wheelchair includes the techniques of obstacle sensing and avoidance, local path navigation, and friendly interactions with users when compared to conventional powered wheelchairs [13]. In order to explain previous research on the friendly interaction, this paper proposed detecting hole during navigation that could contribute to solve the problem better.

Ultrasonic ranging is another technique for distance measurement. The speed of sound travelling over a few meters and reflecting from a solid object gives a kind of delay in milliseconds. This is suitable for the measurement by a hardware timer in microcontroller. The sensor operation start when a short burst of high-frequency sound such as 40 kHz is transmitted and then should finished by the time the reflection return. This is required to avoid the signals being confused by the receiver. To determine the distance to an object, sensors calculate the time interval between sending the signal and receiving the echo.

As shown in Figure 2.2, MaxSonar-EZ1 offers very short to long-range detection and ranging, in an incredibly small package with ultra low power consumption. The MaxSonar-EZ1 detects objects from 0-inches to 254-inches (6.45-meters) and provides sonar range information from 6-inches out to 254-inches with 1-inch resolution. Objects from 0-inches to 6-inches range as 6- inches. The interface output formats included are pulse width output, analog voltage output, and serial digital output.

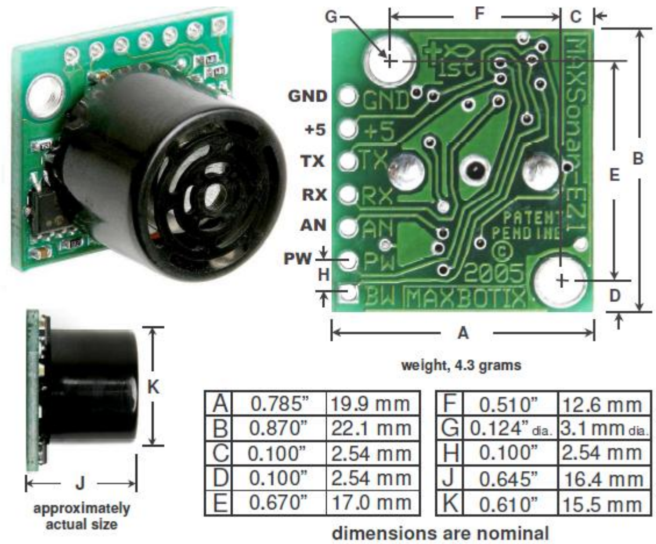

Figure 2.2 *Maxbotic EZ1 Sonar Sensor*

Some of the features have been listed below:

- Continuously variable gain for beam control and side lobe suppression
- Object detection includes zero range object
- Single 5V supply with 2mA typical current draw
- Readings can occur up to every 50mS, (20-Hz rate)
- Free run operation can continually measure and output range information
- Triggered operation provides the range reading as desired
- All interfaces are active simultaneously
- Serial, 0 to 5V
- 9600Baud, 81N
- Analog (10mV/inch)
- Pulse width (147uS/inch)
- Learns ringdown pattern when commanded to start ranging
- Designed for protected indoor environments
- Sensor operates at 42KHz
- High output 10V PP square wave sensor drive

Sample results for measured beam patterns are shown in Figure 2.3 below. The detection pattern is shown for; (A) 0.25-inch diameter dowel, note the very narrow beam for close small objects, (B) 1-inch diameter dowel, dowel, note the long narrow detection pattern,(C) 3.25-inch diameter rod, note the long controlled detection pattern,(D) 11-inch wide board moved left to right with the board parallel to the front sensor face and the sensor stationary. The displayed beam length shows the long range capability of the sensor.

The displayed beam width of (D) is a function of the specular nature of sonar and the shape of the board (i.e. flat mirror like) and should never be confused with actual sensor beam width.

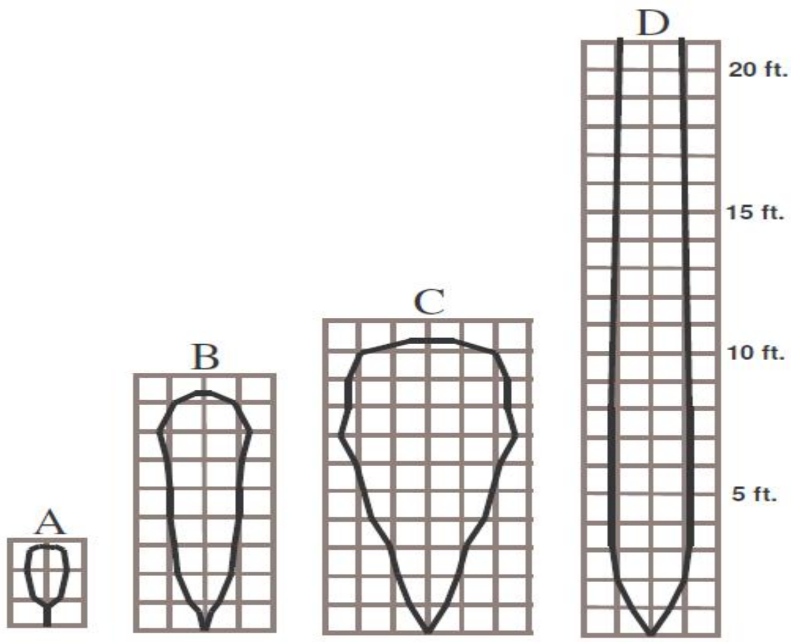

Figure 2.3 The beam pattern on a 12-inch grid.

Based on datasheet of MaxSonar-EZ1, there are three types' connection for I/O port which are analog connection, PWM connection and Bandwidth connection. Each of the connection will have its own timing description in order the sensor to function properly.

For the sensor firstly to operate, 250mS after power-up, the MaxSonar-EZ1 is ready to accept the RX command. If the RX pin is left open or held high, the sensor will first run a calibration cycle (49mS), and then it will take a range reading (49mS). Therefore, the first reading will take 100mS. Subsequent readings will take 49mS. The MaxSonar-EZ1 checks the RX pin at the end of every cycle. Range data can be acquired once every 49mS. Each 49mS period starts by the RX being high or open, after which the MaxSonar-EZ1 sends seven 42 kHz waves after which the pulse width pin (PW pin) is set high. When a target is detected the PW pin is pulled low. The PW pin is high for up to 37.5mS if no target is detected. During the next 4.7mS the serial data is sent. The remainder of the 49mS time is spent adjusting the analog voltage to the correct level. When a long distance is measured immediately after a short distance reading, the analog voltage may not reach the exact level within one read cycle. The MaxSonar-EZ1 timing is factory calibrated to one percent and in use is better than two percent.

Each time after the MaxSonar-EZ1 is powered up, it will calibrate during its first read cycle. The sensor uses this stored information to range a close object. It is important that objects not be close to the sensor during this calibration reading. The best sensitivity is obtained when it is clear for fourteen inches, but good results are common when clear for at least seven inches. If an object is too close during the calibration cycle, the sensor may then ignore objects at that distance The MaxSonar-EZ1 does not use the calibration data to temperature compensate for range, but instead to compensate for the sensor ringdown pattern. If the temperature, humidity, or applied voltage changes during operation, the sensor may require recalibration to reacquire the ringdown pattern. If the temperature increases, the sensor is more likely to have false close readings. If the temperature decreases, the sensor is more likely to have reduced up close sensitivity. To recalibrate the MaxSonar-EZ1, cycle power, then command a read cycle.

2.6 MICROCONTROLLER PIC16F877A

Microcontroller is a small computer on a single integrated circuit which best use to control devices. It contains all the components required for a processor system such as CPU, memory and I/O port. For a complete system to be built, we can use only one MCU chip and few I/O devices such as keypad, buzzer, LCD and other interfacing circuits.

Mainly, it is used in automatically controlled products such as automobile engine control systems, remote controls, office machines, power tools and toys.

Figure 2.5 below show the PIC16F877 pin out that can be seen on the IC package datasheet. This chip can be obtained in three types of different packages such as conventional 40-pin DIP (Dual In-Line Package), square surface mount and lastly the socket format. However, the DIP version is recommended for building a prototype or model.

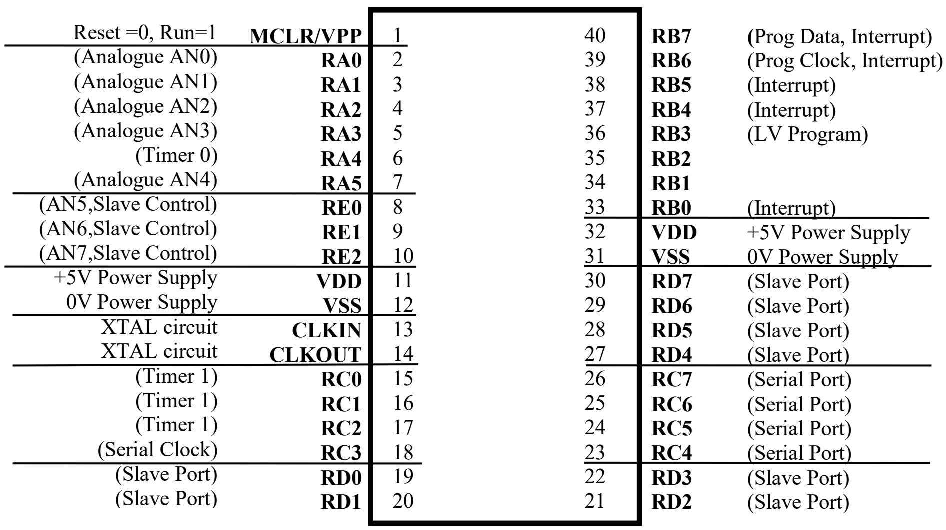

Figure 2.4 PIC16F877 DIP pin out

The MCU can be divided into five ports where most of the pins are for the input and output. Namely the port are A (5), B (8), C (8), D (8) and E (3) and giving total of 32 I/O pins. The pins can all be operated as a simple digital I/O pins, but most have more than one function. The mode of operation is selected by initializing various control registers within the chip. Plus each I/O pin can be programmed as an input or output. The load or current draw that each pin can drive is usually low. If the output is expected to be a heavy load, then it is essential to use a driver chip or transistor buffer.

Ports A and Port E become ANALOGUE INPUTS by default (whether on power up or reset), so they have to set up for digital I/O in this project. The hardware that will connect to these ports are the joystick and sonar range sensor since they produce analog output.

Port B is used for downloading the program to the chip flash ROM (RB6 and RB7), and RB0 and RB4 - RB7 can generate interrupt. Port C gives access to timers and serial ports. Port D can be used as a slave port. Port E providing the control pins for the slave function.

For the power pins (VDD =+5V and VSS=0V), this MCU provide two pairs of pin situated on left and right respectively. The PIC16F877 can actually work down to about 2 V supplies, for the battery and power-saving operation. A low-frequency clock circuit using only a capacitor and resistor to set the frequency can be connected across CLKIN and CLKOUT. MCLR is the reset input which means that when cleared to 0, the MCU stop. When MCLR =1 the system will restart. If an external reset circuit is not connected, this input must be tied HIGH to allow the chip to run. [20]

This MCU also contain its circuitry to generate the system clock. Square wave is generated from the system which indicates the heartbeat of the MCU and all operations are synchronized to it. Obviously, it controls the speed at which the MCU will function. In order to complete the clock circuit, the crystal (20 MHz) or RC components is used. Precisely, to select the operating speed is critical to many applications. To summarize the MCU contains, the following elements is arranged.

i. Instruction set
ii. RAM
iii. ROM,PROM or EPROM
iv. I/O ports
v. Clock generator
vi. Reset function
vii. Watchdog timer
viii. Serial port
ix. Interrupts
x. Timers
xi. Analog-to-digital converters
xii. Digital-to-analog converters

2.7 PIC PROGRAMMER

UIC00A USB ICSP PIC Programmer is one of product from Cytron Technologies Sdn. Bhd that very important in hardware development. The connector for UIC00A (USB ICSP PIC Programmer) is simple and fast method to load program into MCU within less than 5 seconds. It is designed to program Flash PIC MCU which includes PIC12F, PIC16F and PIC18F family. It comes in compact size, powerful and flexible. This programmer obtained it power directly from USB connection, which means no external power supply is required.

On board ICSPTM (In Circuit Serial Programming) connector offers flexible method to load program. It supports on board programming which eliminate the frustration of plug-in and plug-out of PIC MCU. This also allow user to quickly program and debug the source code while the target PIC is on the development board. Figure 2.3 shows the component provided when buying the PIC Programmer.

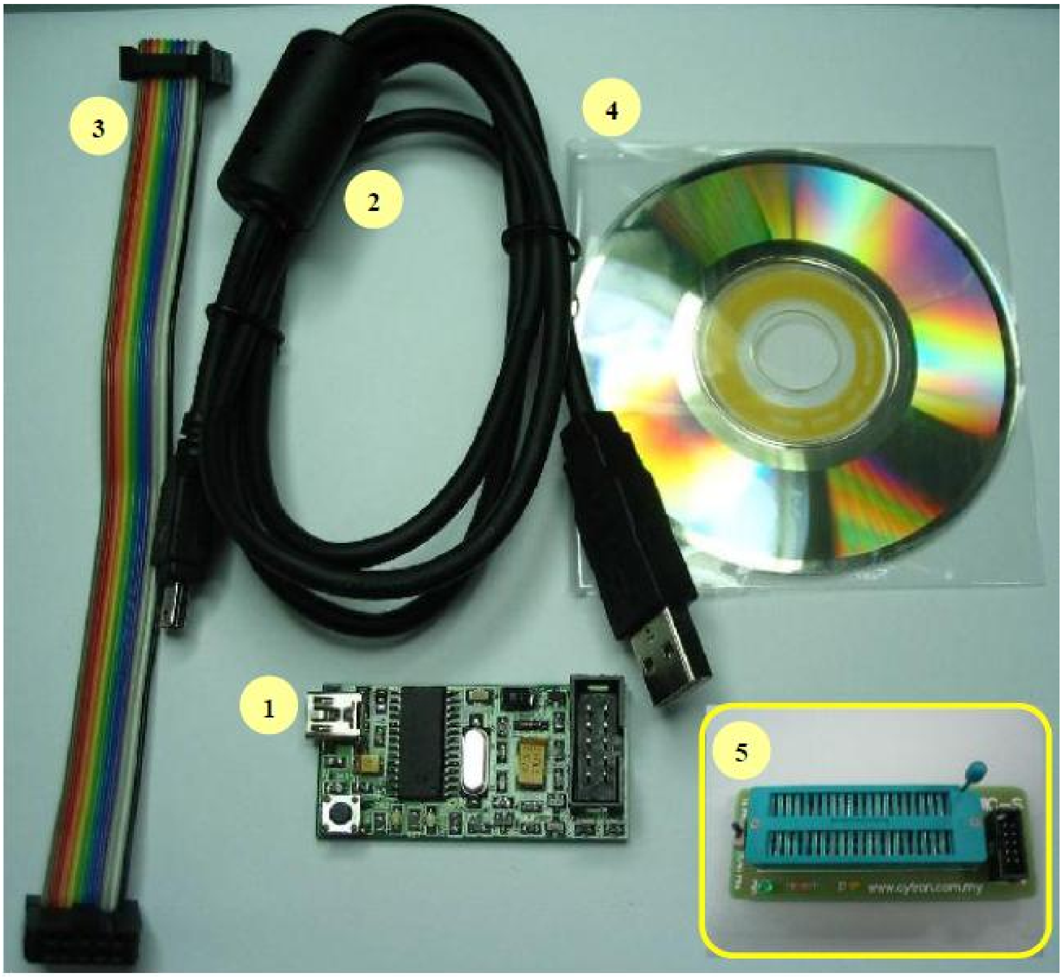

Figure 2.5 The parts and components according to the packing list.

Based on figure 2.5, the parts are:

1. UIC00A main board
2. Mini USB cable
3. Rainbow cable (programming cable)
4. Software Installation and User's Manual CD
5. UIC-S socket board (optional, buy separately from Cytron website)

2.8 MPLAB IDE

MPLAB IDE is a Windows-based Integrated Development Environment for the Microchip Technology Incorporated PIC microcontroller and dsPIC digital signal controller (DSC) families. In the MPLAB IDE, one can:

i. Create source code using the built-in editor.

ii. Assemble, compile and link source code using various language tools. An assembler, linker and librarian come with MPLAB IDE. C compilers are available from Microchip and other third party vendors.

iii. Debug the executable logic by watching program flow with a simulator, such as MPLAB SIM, or in real time with an emulator, such as MPLAB IDE. Third party emulators that work with MPLAB IDE are also available.

iv. Make timing measurements.

v. View variables in Watch windows.

vi. Program firmware into devices with programmers such as PICSTART Plus or PRO MATE II.

2.9 PROTEUS DESIGN SUITE 7

Proteus Design Suite 7 is software which can be divided into ARES Professional 7 and ISIS Professional 7. ARES is used to develop the PCB board while ISIS is used for main purpose of drawing schematic circuit. This software also comes with special features such as analysis the built- in simulation circuit. This feature really helps in hardware debugging which consume time.

For the educational user and engineering author, ISIS excels at producing attractive schematics. It provides total control of drawing appearance in terms of line widths, fill styles, colours and fonts. For a brief summary, some of the functionality in ISIS are described below:

- Runs on Windows 2k and XP.
- Automatic wire routing and dot placement/removal.
- Powerful tools for selecting objects and assigning their properties.
- Total support for buses including component pins, inter-sheet terminals, module ports and wires.
- Bill of Materials and Electrical Rules Check reports.
- Netlist outputs to suit all popular PCB layout tools.
- Hierarchical design with support for parameterized component values on sub-circuits.
- Design Global Annotation allowing multiple instances of a sub-circuit to have different component references.
- Automatic Annotation - the ability to number the components automatically.
- ASCII Data Import - .this facility provides the means to automatically bring component stock codes and costs into ISIS design or library files where they can then be incorporated or even totaled up in the Bill of Materials report.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In this project, microcontroller act like brain to process the input from joystick and sensor before decide the type of navigation for the wheelchair. The block diagram of the system is shown in Figure 3.1.

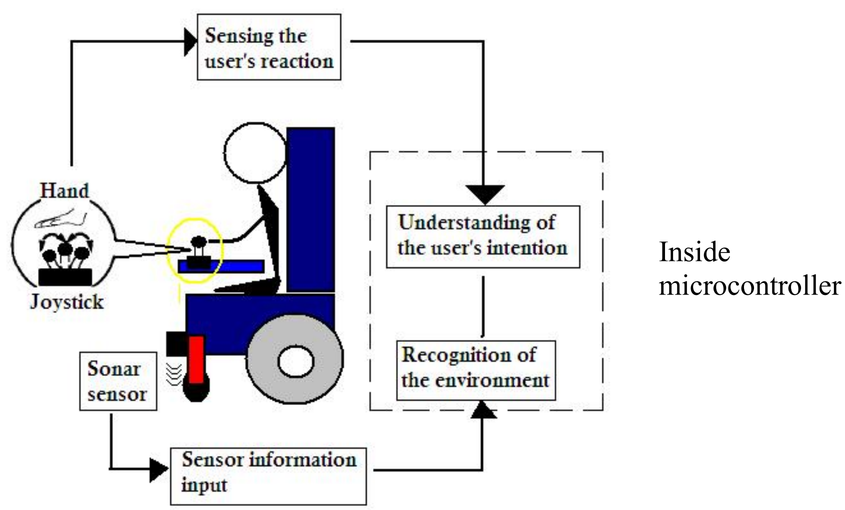

Figure 3.1 The flow of navigation control system

The navigation control system is a closed-loop system. As the user pushes the joystick, microcontroller will update the joystick information for the wheelchair movement. The actual distance from wheelchair bottom to ground will be measured by sensor and feedback to microcontroller. Once the sensor detect hole, the navigation through joystick is no longer use. Microcontroller will instruct motor to avoid the hole by reverse instruction.

Figure 3.2 shows the front view of the project and the side view are depicted in Figure 3.3. The project is divides into two parts that are software and hardware implementation. Each part of the project will discuss in the following section.

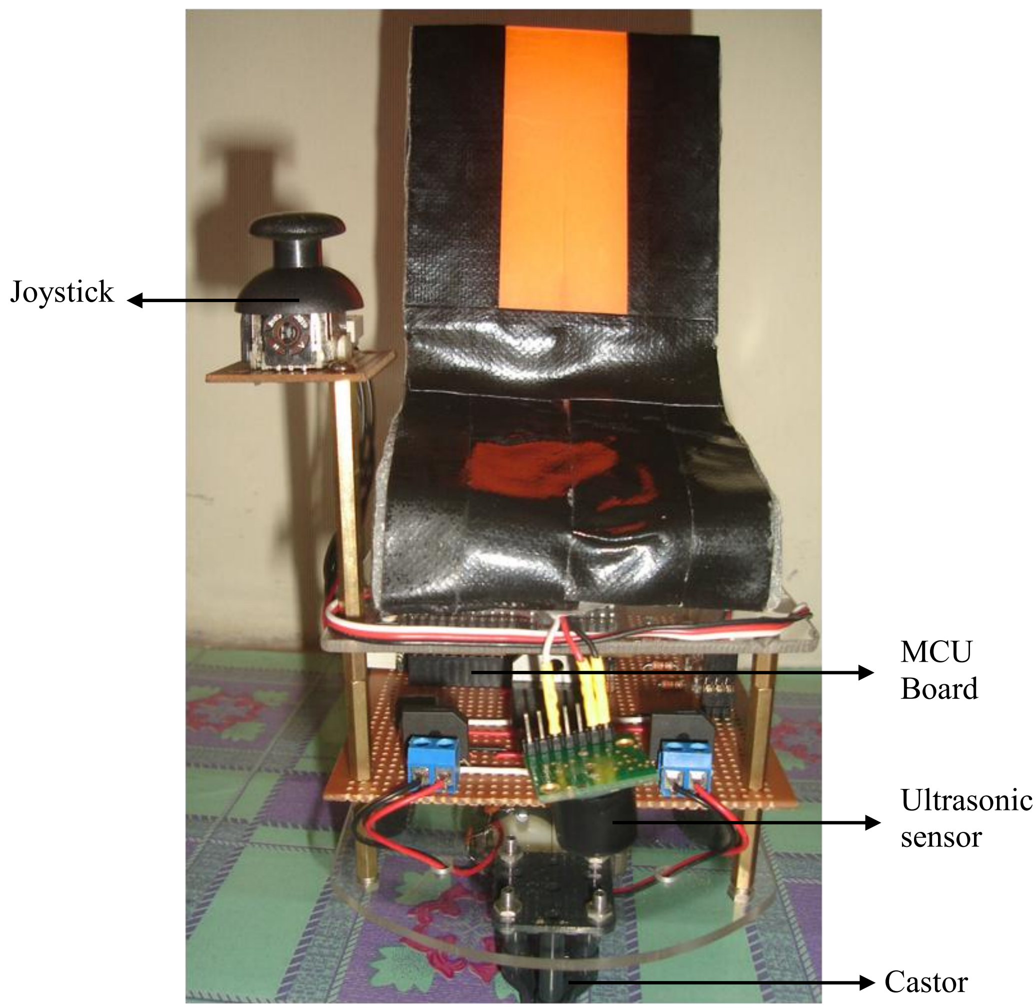

Figure 3.2 Front view of the wheelchair model

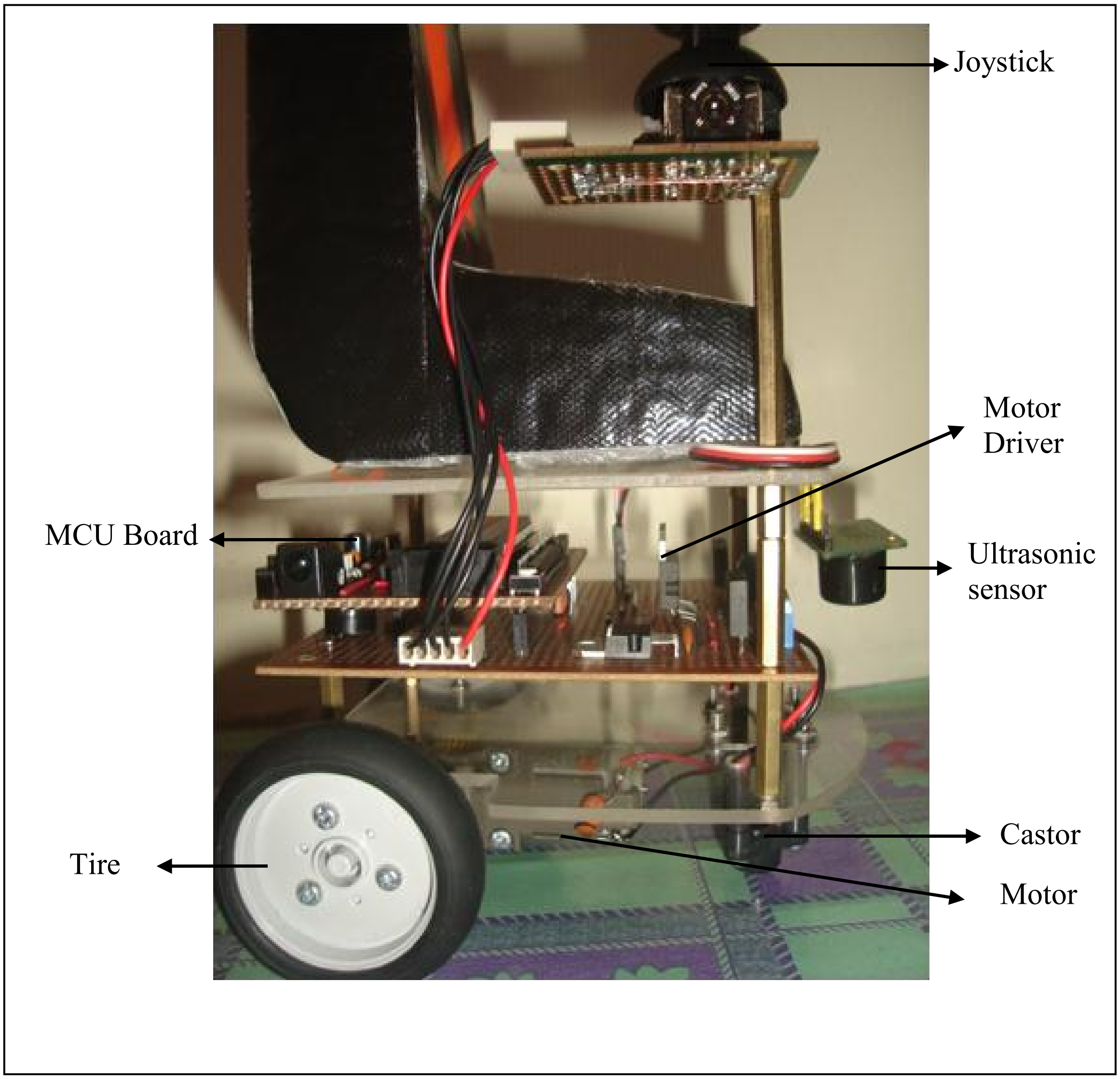

Figure 3.3 Side view of the wheelchair model

3.2 HARDWARE IMPLEMENTATION

The components that had been used are discussed in this section. They are included finger operated joystick, Tamiya Twin Motor Gearbox Kit, L298 Driver Motor, PIC 40-Pins Daughter Board and Maxbotix sonar range sensor.

3.2.1 FINGER OPERATED JOYSTICK

Finger operated joystick comes with small single axis handler. To control, it require minimal effort to move the handles which fit comfortably with finger and thumb operation. These joysticks are suitable for intermittent duty and will not withstand the same abuse as heavy duty joysticks. This joystick is puts on due to the limit in space and it is two-dimensional which mean having two axes of movement. It is configured so that moving the stick left or right signals movement along the X axis, and moving it forward (up) or back (down) signals movement along the Y axis. Figure 3.4 below shows analog outputs with switched reference signals that are proportional to the distance and direction over which the handle is moved. The pin RA1 will receive input from user while pin RA2 will act as the reference voltage. The push button (SW1) is the starting point for wheelchair movement. It is connected with pin RB1 in MCU.

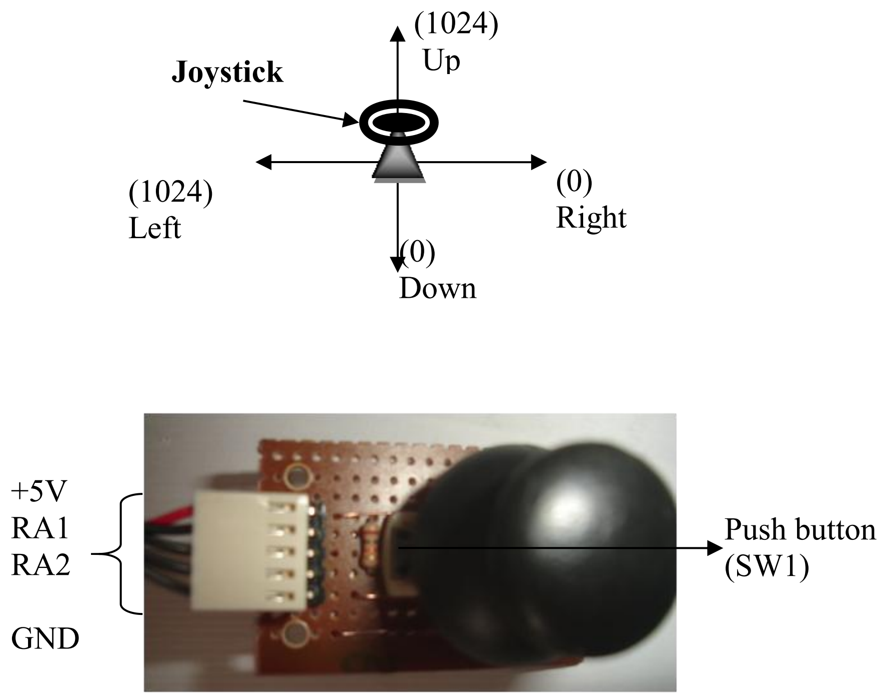

Figure 3.4 Joystick configuration

3.2.2 TAMIYA TWIN MOTOR GEARBOX KIT

The Tamiya twin-motor gearbox consists of two small DC motors that drive separate 3mm hexagonal output shafts. The motor have been built with 58:1 gear ratio in order to provide plenty of power to drive any small robot. Gear ratio can be defined as the relationship between the numbers of teeth on two gears that are meshed. The motor is shown in Figure 3.5.

Figure 3.5 Tamiya Twin Motor Gearbox Kit

The general specifications for the motor that run at typical operating voltage range of 3V to 6V is shown in Table 3.1

Table 3.1 Tamiya Twin Motor Gearbox

Specification	Value
Free-run motor shaft speed	12300 rpm
Free-run current	150 mA
Stall current	2100 mA
Motor shaft stall torque	0.5 oz·in

Based on Table 3.1, the free-run current is the no-load current of the motor when disconnected from the gears in the gearbox. The no-load current of the entire gearbox with the motor connected will be slightly higher and will vary depending on the gear ratio.

3.2.3 L298 DRIVER MOTOR

In order to drive the motor fast, slow, forwards or backwards, L298 driver motor requires two PWM inputs. It has 15 pins in total as in Figure 3.6 which not all of the pins are used in this project. These pins take a PWM input which the frequency of the PWM is dependent upon the motor. Taking 1 KHz as input frequency gives the motor speed to be updates 1 thousand times a second. The duty cycle of the PWM will determine the speed and direction of the motor.

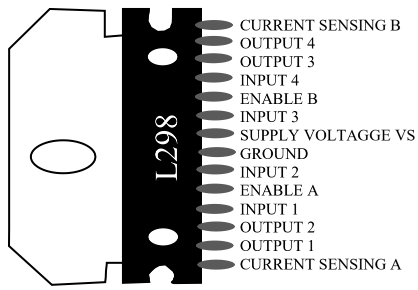

Figure 3.6 Pin out for L298HN Driver Motor

The DC motor drive that will be used in this project is a dual full bridge driver, chip L298. The operating supply voltage of chip L298 is up to 46V and the total DC current up to 4A. Table 3.2 shows function of each pin of chip L298.

Table 3.2 Pin function of chip L298

PIN NUMBER	NAME	FUNCTION
1,15	Sense A, Sense B	Between this pin and ground is connected the sense resistor to control the current of the load
2,3	Out 1,Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	VS	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5,7	Input 1, Input 2	TTL Compatible Inputs of the Bridge A.
6,11	Enable A, Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable

		B).
8	GND	GROUND
9	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10,12	Input 3, Input 4	TTL Compatible Inputs of the Bridge B.
13, 14	Out 3, Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.

3.2.4 PIC 40-PINS DAUGHTER BOARD

The board consists of connector for UIC00A ICSP USB Programmer, toggle switch for power supply, DC power adaptor socket, mini jumper, 5V voltage regulator and 40 pin IC socket for PIC MCU. The details on components are explained on each section. The overview of the board as seen in Figure 3.7

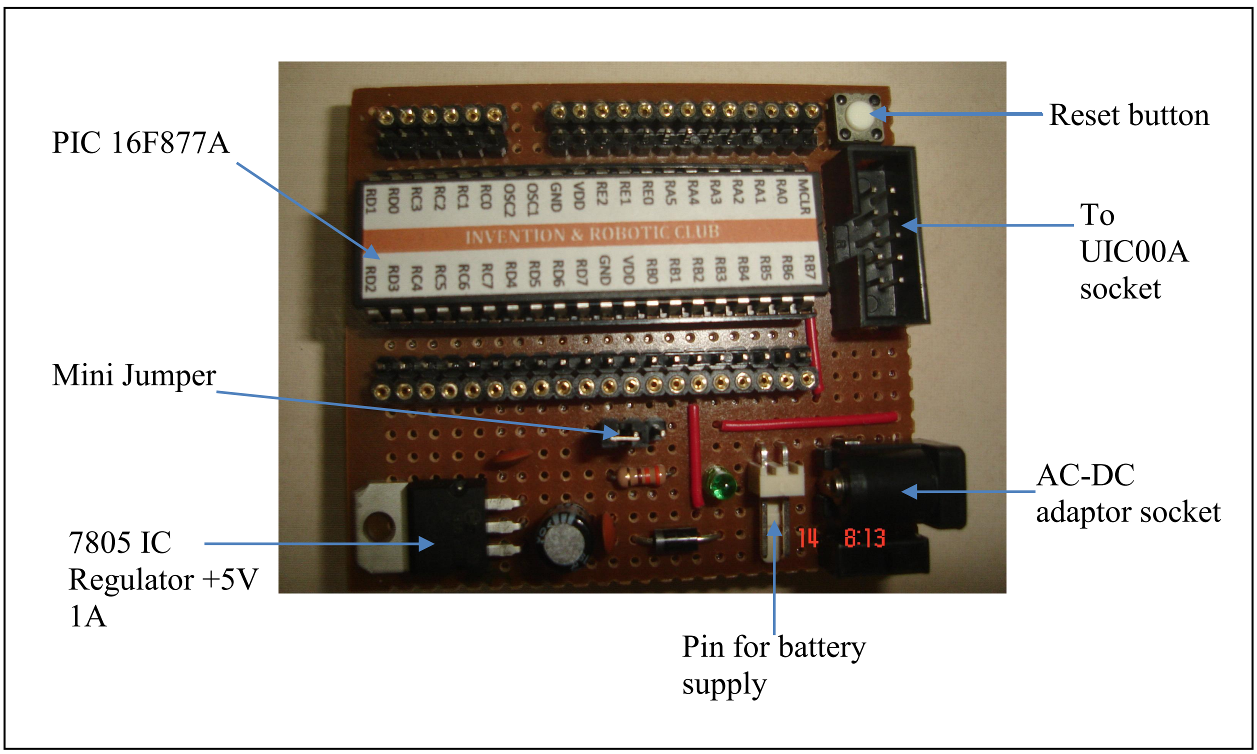

Figure 3.7 40-pins PIC daughter board

3.2.4.1 MICROCONTROLLER PIC16F877

The microcontroller used in this project is PIC16F877A which work as its main brain in moving the motor. Among the features that this PIC has are I/O Ports, Timer, PWM Module, Master Synchronous Serial Port (MSSP) Module, Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART), Analog-to-Digital Converter (A/D) Module and Comparator Module.

PIC16F877A MCU used to control software for the powered wheelchair to function properly. Its embedded circuit consisting of a relatively simple CPU combined with support functions such as a crystal oscillator, timers, watchdog timer, serial and analog I/O. These MCU operate at 20MHz clock rate frequency. This is fair enough for many typical applications which also enabling low power consumption as low as in milliwatts. Generally, they have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption may be just nanowatts, making many of them well suited for long lasting battery applications.

3.2.4.2 POWER SUPPLY

The board was equipped with 5V voltage regulator by using LM7805 chip. This supply is enough for the MCU to operate in normal condition. The voltage regulator circuit also can take supply either from AC-DC Adaptor or battery. The supply voltage from AC-DC adaptor should be ranged from 6V to 9V.

3.2.4.3 UIC00A PROGRAMMING PORT

The board was added up with UIC00A ICSP USB programmer connector which a powerful tool used to download the program into MCU through free-programming-software, MPLAB. The connector is very convenience because it offers simple way for transferring program and save plenty of development time.

3.2.4.4 MINI JUMPER

The mini jumper is functions as a turn pin to supply the MCU with secondary power when using USB port. This means the PC or laptop itself that supply voltage during downloading the program into MCU. Thus, the external power from AC-DC adaptor may not be needed during the transferring.

3.2.5 ULTRASONIC SENSOR

The sensor connection for detection purpose is shown in Figure 3.8 below. The pin “AN” is used because this output is very easy to use which the distance measurement is based on 10mV per inch.

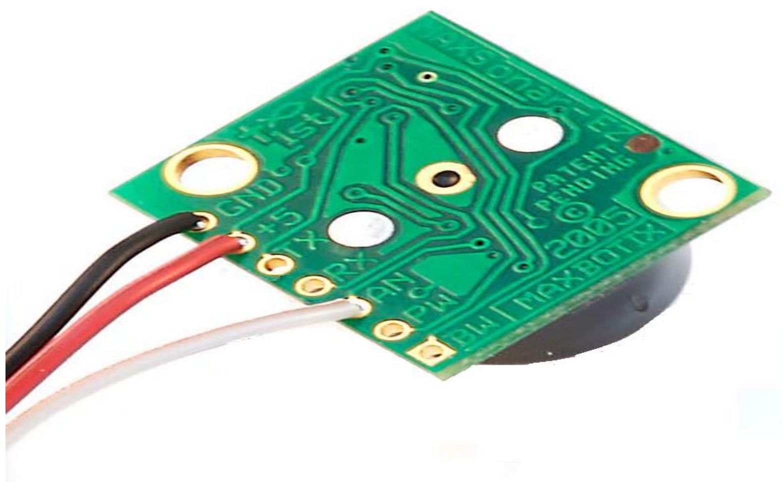

Figure 3.8 The connection of ultrasonic

In order to able verify the operation, the sonar sensor is supply with 5V. By using a microcontroller, it takes only one pin (RA2) to monitor any of the analog outputs. Since only one sensor is used in this project, the circuitry is not complicated. This connection also will continuously update the analog voltage output with 10mV representing 1 inch.

3.3 SOFTWARE IMPLEMENTATION

Program or software implemented in this project was based on C programming. MPLAB IDE is the tool responsible to program the microcontroller operation. With the schematic that has been drawn in Proteus 7, the configuration of the port become easily.

3.3.1 PULSE-WIDTH-MODULATION (PWM) IN MICROCONTROLLER

The Pulse-Width-Modulation (PWM) in microcontroller is used to control duty cycle of DC motor drive. PWM is an entirely different approach to controlling the speed of a DC motor. Power is supplied to the motor in square wave of constant voltage but varying pulse-width or duty cycle. Duty cycle refers to the percentage of one cycle during which duty cycle of a continuous train of pulses. Since the frequency is held constant while the on-off time is varied, the duty cycle of PWM is determined by the pulse width. Thus the power increases duty cycle in PWM.

The expression of duty cycle is determined by,

$$\% \, Duty\ cycle = \frac{t_{on}}{T} \times 100 \tag{3.1}$$

Basically, the speed of a DC motor is a function of the input power and drive characteristics. While the area under an input pulse width train is measure of the average power available from such an input.

3.3.2 ALGORITHM AND PROGRAMMING IN MPLAB

An algorithm has to be developed to make the microcontroller to read the input and respond accordingly. Microcontroller acts as brain of the whole navigation control system. It will receive the desired command from user through joystick. The analog output from joystick will change into digital for MCU to be able process the command. Then, MCU instruct motor to move according to the processed command. In the mean time, the sonar sensor will update its analog data and send to MCU. Once again, ADC will occur. The actual distance will be compared with the fixed distance in MCU. When the actual distance is exceed the fixed distance, buzzer will turn ON and the wheelchair will move backward in high speed automatically. This indicates the application to detect hole during the user moving the wheelchair.

Therefore, the algorithm is established and represented by a flowchart in Figure 3.9 below.

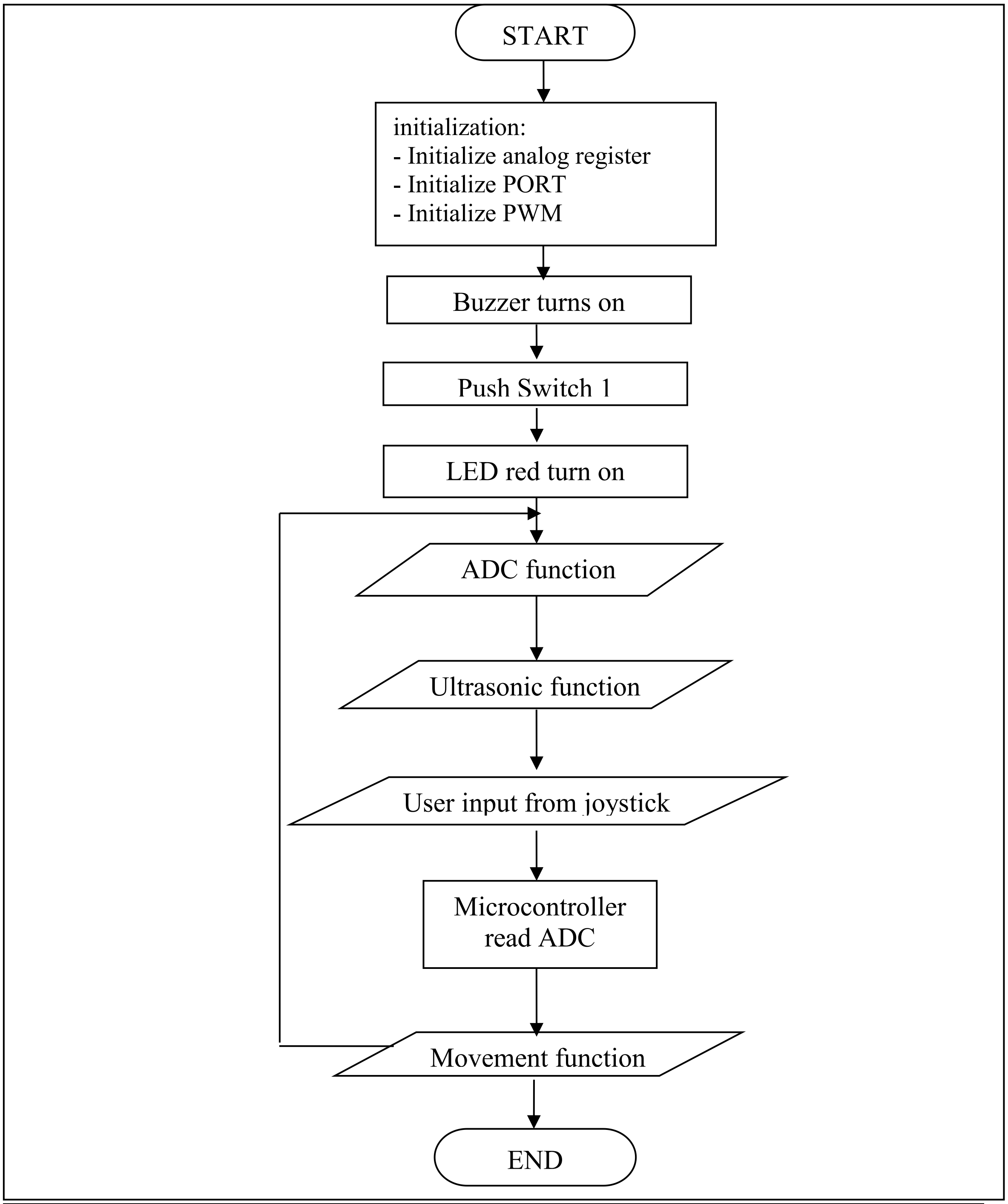

Figure 3.9 Flowchart of main operation

This flowchart is then translated into C language and compiled using MPLAB, the PIC16F877A software development tool. The overall program can be referred in Appendix 5.

3.3.3 PROCESSING EXPLANATION OF PROGRAM

There are two parts of the program which are main program and interrupt program. The microcontroller will always run the main program until there is an interrupt occurred. The interrupt inputs receive from the ultrasonic sensor. When microcontroller receives an interrupt flag, then it will jump to interrupt process.

a) Initialization of port A to E

The process of initialize I/O and analog ports begins. Port A (RA0 – RA7), Port B (RB0) and Port E (RE0 – RE2) are set as input while other ports are set as output from MCU.

b) Initialization of PWM

The PWM signal that generate will input into the motor driver using the PORTC and PORTD on the MCU. A PWM output as seen from Figure 3.10 has a time-base (period) and a time (duty cycle) that the output stays high. To calculate the frequency of the PWM, do inverse of the period (1/period).

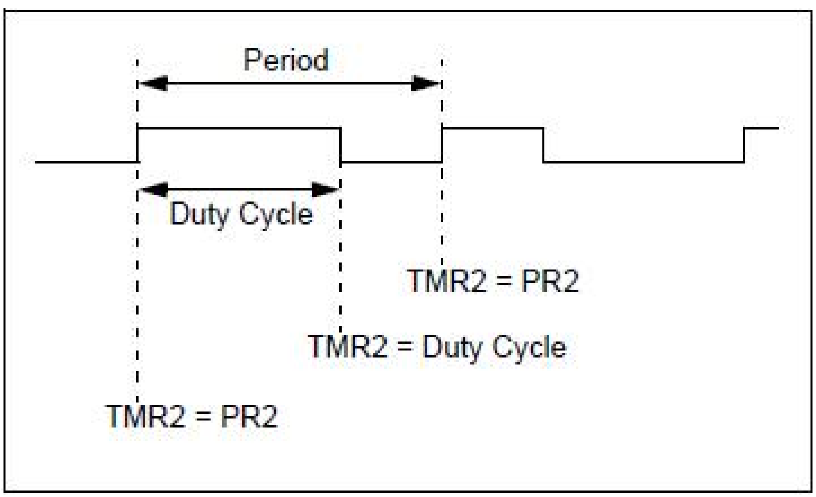

Figure 3.10 The PWM output

To setup for PWM operation, the following steps should be taken when configuring the CCP module for PWM operation:

i. Set the PWM period by writing to the PR2 register as PR2 =255
ii. Set the PWM duty cycle by writing to the CCPR2L register and CCP2CON<5:4> bits
iii. Make the CCP1 pin an output by clearing the TRISC<2> bit.
iv. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
v. Configure the CCP2 module for PWM operation.

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

$$PWM\ period = [\ (PR2) + 1] \times 4 \times T_{osc} \times (TMR2\ prescale\ value) \qquad (3.2)$$

In this project, PR2 is set to 255 because it is the maximum value of 8 bit for maximum range. So, the PWM period is then become

PWM period = (255+1)* 4 * (1/20M) * (1)
= 51.2µs
= 19.53kHz

where Fosc = 20MHz and TMR2 prescale =1

The PWM duty cycle is specified by writing to the CCPR2L register and to the CCP2CON<5:4> bits. Up to 10-bit resolution is available. The CCPR2L contains eight MSbs and the CCP2CON<5:4> contains two LSbs. This 10-bit value is represented by CCPR2L:CCP2CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

$$PWM\ duty\ cycle = (CCPR2L:CCP2CON<5:4>) * T_{OSC} * (TMR2\ prescale\ value) \qquad (3.3)$$

c) ADC Function

The flow of ADC operation is shown in Figure 3.11. The conversion of an analog input wills results 10-bit digital number to be produced. The A/D module has high and low-voltage reference input that is software selectable. The A/D module has four registers and namely they are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

After the conversion of analog input signal has completed, the result will be arranged in ADRESH and ADRESL respectively. The ADCON0 register controls the operation of the A/D module. While the ADCON1 register configures the functions of the port pins. In this project, the port pins configured as analog inputs are RA1 (joystick), RA2 (sensor), and RA3 as the voltage reference.

To summarize the ADC function in this project, the steps below is arranged:

1. Configure the A/D module:
    - Configure analog pins/voltage reference and digital I/O (ADCON1)
    - Select A/D input channel (ADCON0)
    - Select A/D conversion clock (ADCON0)
    - Turn on A/D module (ADCON0)
2. Wait the required acquisition time.
3. Start conversion:
    - Set GO/DONE bit (ADCON0)
4. Wait for A/D conversion to complete by either:
    - Polling for the GO/DONE bit to be cleared (interrupts disabled); OR
    - Waiting for the A/D interrupt
5. Read A/D Result register pair (ADRESH:ADRESL).

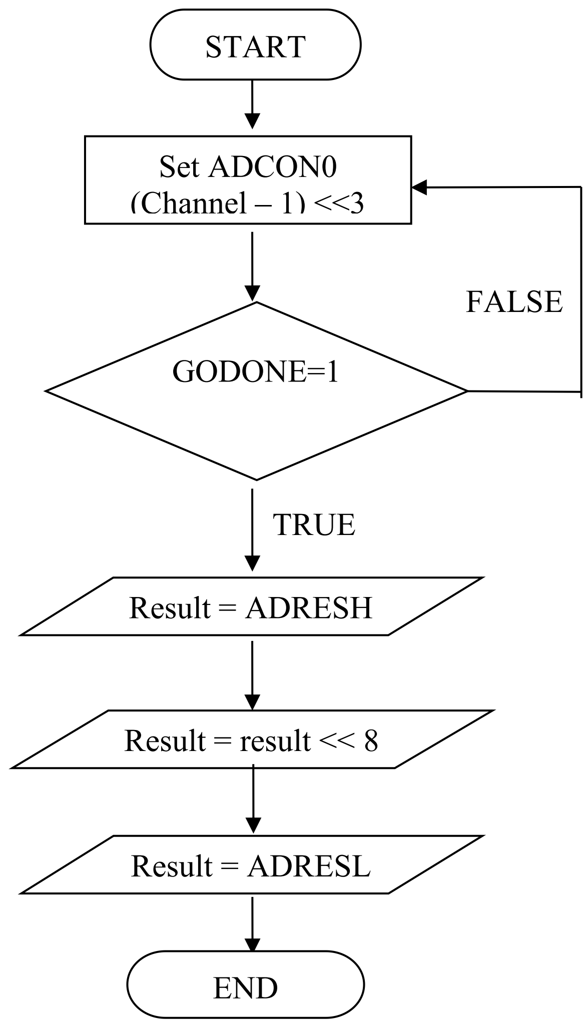

Figure 3.11 Flowchart of ADC function

d) Ultrasonic Function

The interruption of ultrasonic sensor in the program occur when it detect holes. The flowchart of hole detection and its consequence in programming is shown in Figure 3.12 below. The program start once the supply voltage is turn ON. Then, the process of initialization of ADC results as “distance”. The programs proceed by comparing the “distance” whether the value has exceeded the fixed distance limit or not. When the “distance” exceed, the navigation is taken over by the MCU. It will command the backward operation and will sound the buzzer. If “distance” not exceed, the ADC will continue its operation to update the “distance” value.

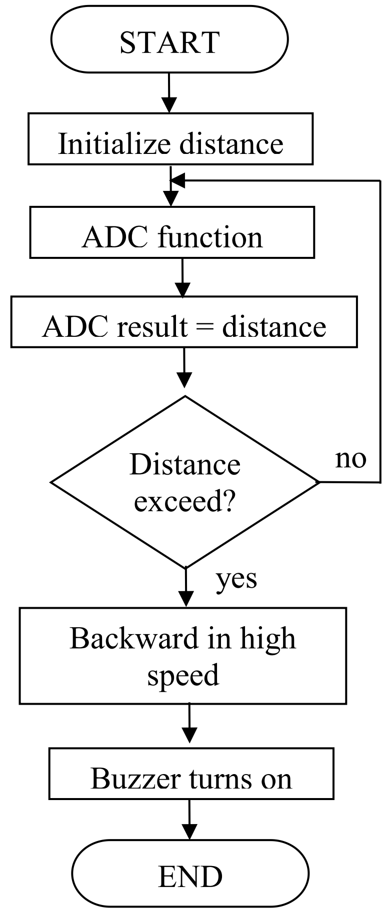

Figure 3.12 Flowchart of Ultrasonic function

e) User input from joystick

The wheelchair has one forward speed set points and one reverse speed set point. As user move joystick, microcontroller read and analyzes the analog input. The outputs from microcontroller indicate as the input for each motor to move either forward or backward. Figure 3.13 shows the flow of wheelchair navigation by user. Different LED color that turn ON indicate the type of command that user propose.

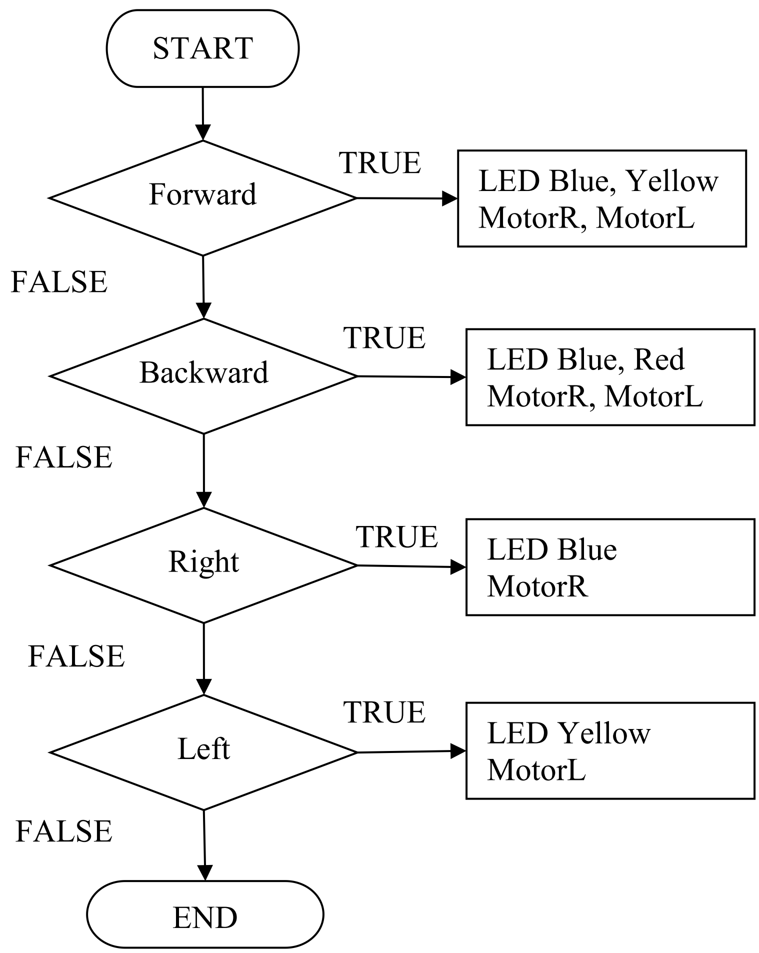

Figure 3.13 Flowchart of Movement function

Motor provide the tires to propel in wheelchair movement. To achieve different types of operation to the tires, specific command must be followed. By referring to the chosen motor driver spec, Table 3.3 is come out as shown below.

Table 3.3 Motor main operation.

Signal on				Result/action
LA	LB	RA	RB	
0	1	1	0	Forward
1	0	0	1	Backward
0	1	0	0	Turn Right
0	0	1	0	Turn Left
1	1	1	1	Motor Stop

The software implementation of the L298 drive motors occur at port pins RD0, RD1, RC0 and RC3 respectively. The motor driver spec can be referred to Table 3.2 above.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 INTRODUCTION

Some experiments and observation had been conducted for the project. An experiment to find out the relationship between voltage supply and motor speed had been done. Then, data collection had done at each speed for DC motor speed control system to observe performance of the system. Last but not least, analysis on the navigation system is made.

4.2 EXPERIMENT: DETERMINE RELATIONSHIP OF LOAD AND MOTOR SPEED

An experiment is conducted to determine the relationship between voltage supply and speed. The procedures and the result will be discussed in following sections.

4.2.1 PROCEDURES

1. The circuit was supplied with 9V using AC-DC adaptor and the exact load was attached onto the wheelchair model
2. Joystick has been push forward to let it move straight on the 60 cm long path.
3. The stopwatch started as the wheelchair model move.
4. The time taken to move along the path was recorded in Table 4.1.

5. The step 2 to step 4 was repeated five times to find the average time and was recorded in Table 4.1.
6. The experiment continued with different weight of load to be attached to the wheelchair model. The load was changed based on Table 4.1.
7. A graph of weight versus time was plotted. Based on the graph, the maximum load the wheelchair model can bring has been found.

4.2.2 EXPERIMENT RESULT ANALYSIS

From experiment, the data was recorded in Table 4.1 and the graph of weight versus time is shown in Figure 4.1.

Table 4.1 Relationship of weight and time

Weight (kg)	Time(s)	Average Time (s)
0.20	7.35 7.20 7.30 6.76 7.55	7.232
0.35	7.96 7.73 7.83 8.66 8.54	8.144
0.55	10.60 10.59 10.10 10.55 10.60	10.488
1.00	12.48 11.48 13.04 14.55 14.19	13.148

Figure 4.1 shows the experimental results of the time taken for the wheelchair to carry load with a constant of 9V supply. The wheelchair model can carry up to 1 kilogram load during move forward. The speed is accordance to the maximum load carried by the

wheelchair model. The speed performance becomes poor as more load attached to the wheelchair. By taking this result, it suggests that every 0.2kg load, the speed decrease by $0.01ms^{-1}$.

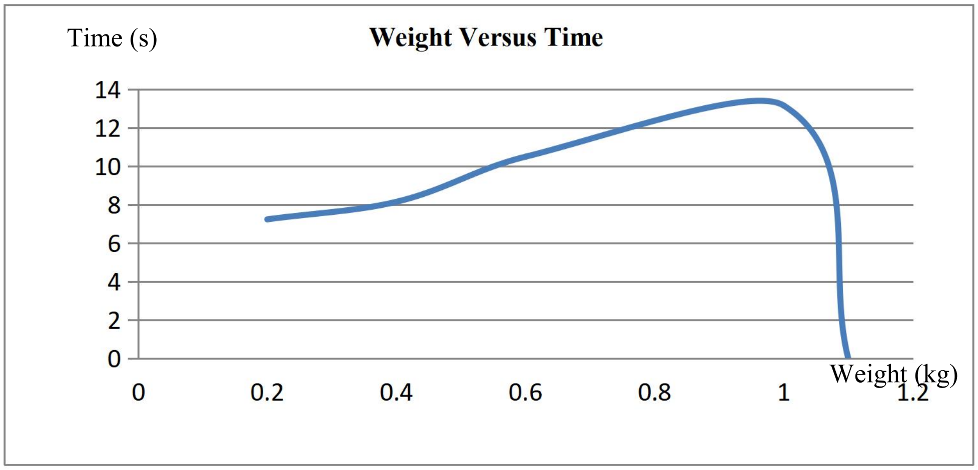

Figure 4.1 Time measurement based on varies load

4.3 TURNING ANALYSIS

The three-wheels provide a good turning on rough terrain such as on the road surface. The model also can turn to left and right very well although the load its carry exceed the maximum load of 1 kilogram.

4.4 RANGE SENSOR DETECTION

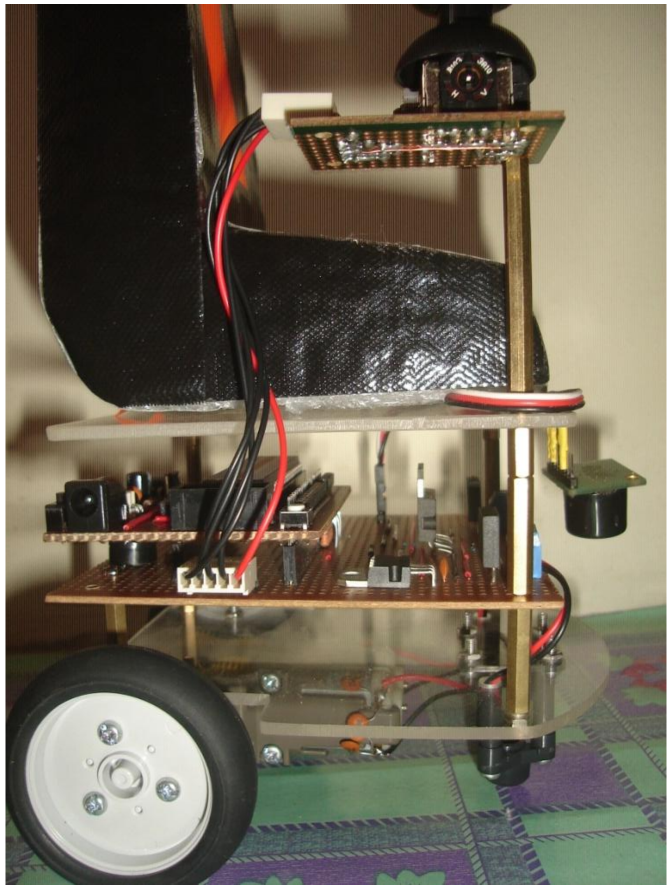

Figure 4.2 The location for the sensor to detect hole along the path

The sensor is put 3 inch above the ground as shown in Figure 4.2. Some observations during the sensor operation has tabulated in Table 4.2

Table 4.2 The observation analysis for the sensor

ACTION	OBSERVATIONS
Put the sensor vertical to the ground	No operation occurs. Expecting the wheelchair will move backward with buzzer turn ON
Put the sensor inclined high about 30º	The motor slowly move backward with buzzer turn ON
Put sensor facing a long	As the sensor slowly facing the far object, motor increase

range object	its speed gradually
Moving forward	Sensor interrupts the process sometimes not regarding the specific distance. The interruption makes the wheelchair move backward, not continuously but in three steps. When changing the program, buzzer sometime not sounded when sensor was in operation.

The consequence based from the observation is then discussed. First and foremost, each time after the sonar sensor is powered up, it will calibrate during its first read cycle. Note that the sensor will use this stored information to range a close object. It is important that objects not be close to the sensor during this calibration cycle. It is stated that the best sensitivity for calibration is obtained when it is about fourteen inches. However, due to small size model that had been built this is hardly to achieve.

In addition of the calibration process, if an object is too close facing the sensor, the sensor may also then ignore objects at that distance. This means that the sensor acting like no object in front of it. Plus, the sensor also have high sensitivity which if the temperature increases, the sensor is more likely to have false close readings.

Furthermore, the sensor has some noise cancelling which made the sensor unable to trace the actual distance. As illustrate in Figure 4.3, the sensor will receive two echoes at once.

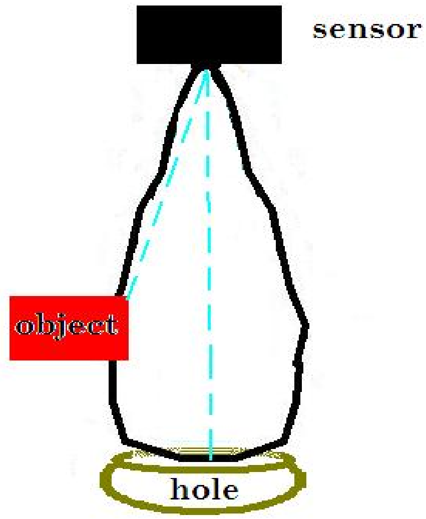

Figure 4.3 Illustration on single sensor with multi-return path

The object in the Figure 4.3 can be anything including the very previous transmitting signal. Let say, the sensor is mounted 12 inches off the floor. Then, pull back the wheelchair from the wall until no return is detected. The maximum range that occurs is at 3 meters and not at the 6 meters as expected. But if the wheelchair is pulled back another 0.5 meters to 3.5 meters, the signal reappears and the sensor detects the wall past 6 meters. What happened is that there were two signal return paths back to the sensor. It can be namely the direct path (expected) and the indirect path (the unexpected one that included the floor refection). At this distance and with this setup, phase cancellation can occur at the sensor. So the sensor does not have a signal to detect, not because the sensor is detective, but because of multi path signal cancellation from phase effects. However, the effects are well defined in wave theory, but it can be vary in real world.

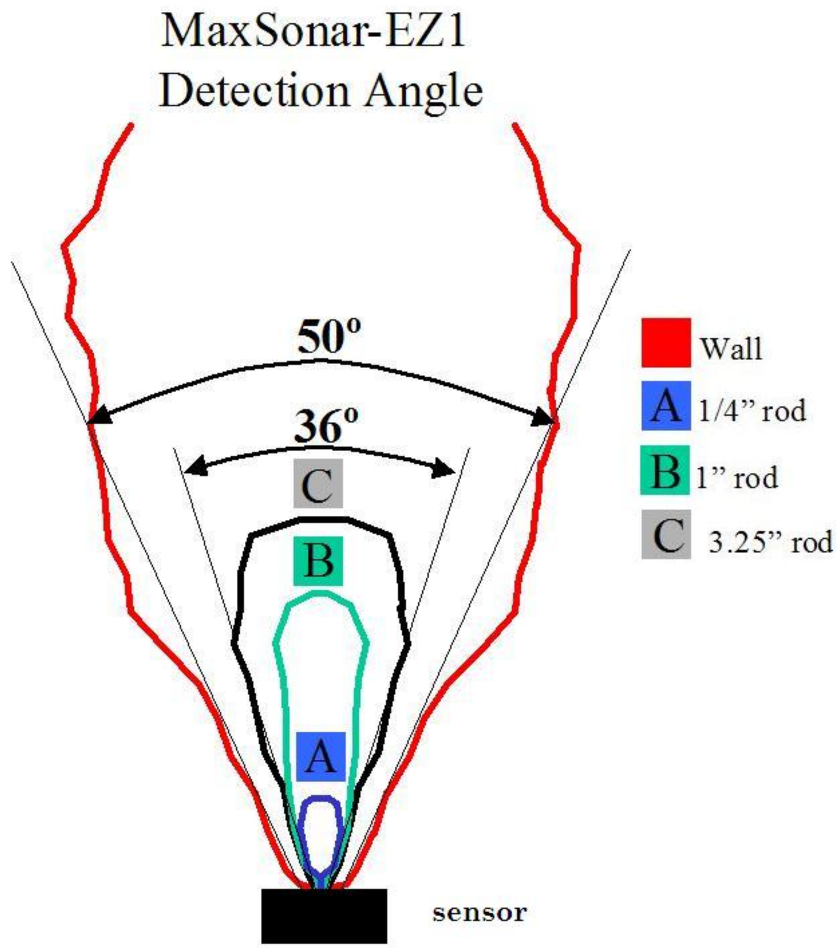

Figure 4.4 The detection angle in the sensor

Based on Figure 4.4, sensor has different type of detection angle for different range of distance such as detection at 0.25 inch, 1 inch and 3.25 inch. These suggest if there is electrical noise introduced on the line, it can cause the sensor to output unstable readings. Among the known items that cause excessive power supply noise are Sharp infrared range sensors, XBee radios, some wireless control systems, some switching power supplies and some servos. .

4.5 OTHERS OBSERVATION

The following test has been set up to evaluate the performance of the wheelchair.

The result was based on AC-DC adapter with maximum of 500mA to supply the wheelchair model. The motor did not move at all when supplied with 6V and lower. Thus, the starting supply was at 7.5V. Based on Figure 4.5, the model took 15.6 seconds to move in 1 meter at 7.5V supply. At 9V supply, it took 9.2 second to move on same distance. The wheelchair model took shorter time when supplied at 9V, thus this shows as the voltage increase the speed also increases.

The motor not moving at starting voltage of 6V because of the motor need 6V supply while the PIC board need 5V supply. Given 6V supply will not enough to support both components. Based from reading, the Pololu motor itself has been tested and lead to lower the voltage supplied will longer the motor lifespan. Thus, for the motor to work in longer time it is recommended at 7.5V.

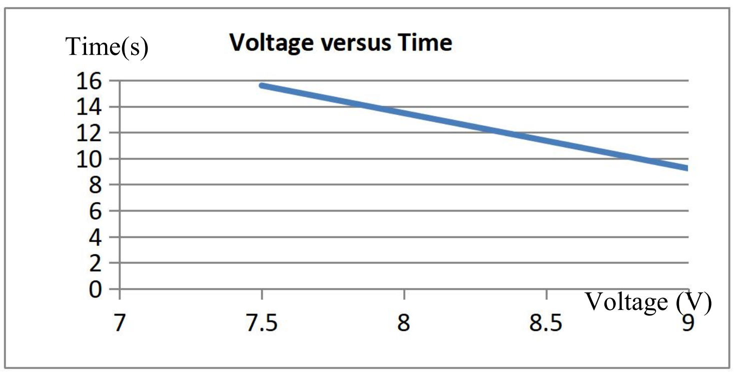

Figure 4.5 Time measurement based on varies supply

Based on observation, using motor in a long time caused the motor to move slowly. This might due to using high supply which then caused damage to the brushes in motor. Unfortunately, since these are small 'toy' motors, the manufacturer really doesn't offer any cycle-life numbers, even when operating the motor at the recommended 3V or less.

There also another condition involving rolling friction that can be discussed. The resistance encountered when tires are moved in contact on a surface can be called as rolling friction. Rolling friction can be considered as the load on the wheelchair and it depends on the type of tire and road surface. The rolling friction for a specific road can be calculated by measuring the time taken by wheelchair to travel the known length at constant speed. The input power is also measured for the same time. Since the surface is flat power remains constant for whole length.

The wheelchair was observed to move along in a straight line as in Figure 4.6. However, its front caster tend to follow the rail on the floor as in Figure 4.7

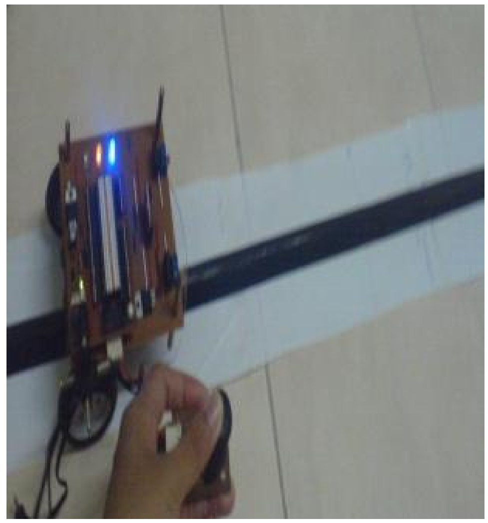

Figure 4.6 Wheelchair move in straight line

Figure 4.7 The caster tends to follow the rail

During the experiment set up the speed for left-motor and right-motor was not synchronize especially for wheelchair to move forward and backward. It was found that the right-motor move faster during forward operation. While during backward, the left-motor move faster.

CHAPTER 5

CONCLUSIONS

5.1 CONCLUSION ON PROJECT

This project has been successful establish joystick in movement of EPW model. In addition, the navigation is aided by sonar range sensor which successfully installed. Many difficulties related to navigation system have been overcome in software programming. Moreover, this system controller is responsible for generating the PWM signals that will be delivered to the motor and insuring it move at only one speed point.

Three wheels was implemented and found working satisfactory for navigation. The efficiency is depends on the load. At lower load, it is possible to get still higher efficiency. The wheelchair model can bring 1 Kg load and travel in $0.0456\, ms^{-1}$ with 500 mA battery and this range could be further increased by adding battery in parallel. The effort made by rider is reduced considerably as compare to manual wheelchair.
Motor will be overheated which can be caused by excessive stalling, even at very low voltages. Thus, it is important to know the exact current of the motor so that it has long lifespan.

The performance of the model indicates there is a cautious possibility today to build a functional intelligent wheelchair that is practical and helpful to people with certain types and degrees of handicap.

5.2 PROBLEMS ENCOUNTER

During background study, some implementation of sip-and-puff navigation system to the wheelchair has been discussed. However, it is hardly to achieve due to the suitable pressure sensor itself was not founded and this consuming the time. Therefore, to keep on track, the navigation chosen was based on joystick. Then, if there still time, the navigation system will be improved using depth sensor which was achieved at last.

Forward and backward operations are not properly function. The right-wheel move faster at forward operation whiles the left-wheel move faster at backward operation. The problem was encountered by changing the PWM value for each operation.

The front castor tends to follow long-rail or mark on the road. This will make the wheelchair not moving by the user command. To make the wheelchair moving out of the mark, simply do the turning command. The front caster moving along rail will be added as disadvantage of the design that could be overcome in the future. Thus, the sonar sensor is put at the front castor. The sensor will be used to allow only specific hole for castor to be able move. In addition, the sensor will provide a safe trip to the user.

In this project, there are three times ADC occurs. The first conversion is for the ultrasonic sensor while the other two conversions are for the joystick. Therefore, this will make delay in system operation. The crucial is when the sensor detects range, it will late in updating the latest data. This results the sensor sometimes late to detect holes during the navigation.

5.3 FUTURE RECOMMENDATION

The L298HN motor driver has advantages in dc motor control. For examples, it accepts PWM input, control two dc motors simultaneously and also capable controlling stepper motors for a future means. The motor driver also can be used into the real wheelchair, which the motor will have high torque and take up to 24V supply voltage. In addition, the range speed of motor can be modified using the driver.

The test should be more variety such as jerkiness test, collision risk test, rough terrain test and smooth hallway test course in order to improve the wheelchair navigation system in the future.

The ultrasonic sensor technology can also be used for:

- Wind speed and direction (anemometer). For measuring speed or direction, multiple detectors are used and calculate the speed from the relative distances to particulates in the air or water.
- Fullness of a tank and speed through air or water. To measure the amount of liquid in a tank, the sensor measures the distance to the surface of the fluid.
- medical ultrasonography
- burglar alarms
- Non-destructive testing.

For the ultrasonic sensor, the serial digital output and the pulse width outputs are taken directly from the time of flight measurement. They will have no additional noise present on them and will be the most accurate.

The programming in this project should be improved in order for users' convenient in using joystick. This because of the controllers should be customized to individual users as the maximum average speed varies dependant on the operator [12].

REFFERENCES

[1]Yoshinori Kuno, Satoru Nakanishi, Teruhisa Murashima, Nobutaka Shimada and Yoshiaki Shirai, "*Robotic Wheelchair with Three Control Modes*", Proceedings of the 1999 IEEE International Conference on Robotics & Automation Demit, Michigan, May 1999

[2] R. Rahulanker and V. Ramanarayanan "*Battery Assisted Wheel Chair*" presented at Proceedings of India International Conference on Power Electronics 2006

[3] Dan Ding, Rory A. Cooper, Songfeng Guo, and Thomas A. Corfman ,"*Analysis of Driving Backward in an Electric-Powered Wheelchair*", IEEE Transactions On Control Systems Technology, Vol. 12, No. 6, November 2004

[4] Salmiah Ahmad and M. O. Tokhi, "*Forward and Backward Motion Control of Wheelchair on Two Wheels*" presented for Department of Automatic Control and Systems Engineering,The University of Sheffield, UK

[5] Bell, D., Borenstein, J, Levine, S., Koren, Y., and Jaros, A., *"The NavChair: An Assistive Navigation System for Wheelchairs, based on Mobile Robot Obstacle Avoidance."* proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, May 8-13, 1994, pp. 2012-2017.

[6] Levine, S.P., Bell, D.A., Jaros, L.A., Simpson, R.C., Koren, Y., and Borenstein, J., *"The NavChair Assistive Wheelchair Navigation System."* from IEEE Transactions on Rehabilitation Engineering, Vol. 7, N0. 4, December 1999, pp. 443-451.

[7] Ye, C. and Borenstein, J., *"A Method for Mobile Robot Navigation on Rough Terrain."* proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, April 26-May 1, 2004, pp. 3863-3869.

[8] Ojeda, L., Borenstein, J., *"Non-GPS Navigation for Security Personnel and Emergency Responders."* Journal of Navigation. Vol. 60 No. 3, September 2007, pp. 391-407.

[9] Takashi Gomi **"***The TAO Project: Intelligent wheelchairs for the handicapped"* from AAAI Technical Report FS-96-05. (1996)

[10] Qiang Zeng, Chee Leong Teo, Brice Rebsamen, and Etienne Burdet*,* Member, IEEE,*" A Collaborative Wheelchair System",* from IEEE Transactions On Neural Systems And Rehabilitation Engineering, Vol. 16, No. 2, April 2008

[11] Rory A. Cooper, Thomas A. Corfman, Shirley G. Fitzgerald, Michael L. Boninger, Donald M. Spaeth, William Ammer, and Julianna Arva, "*Performance Assessment of a Pushrim-Activated Power-Assisted Wheelchair Control System",* IEEE Transactions On Control Systems Technology, Vol. 10, No. 1, January 2002

[12] Yoshihiko Takahashi and Hirofumi Takagi, "*Joy Stick Operation Speed of Intelligent Robotic Wheelchair"*, International Conference on Control, Automation and Systems 2007, Oct. 17-20, 2007 in COEX, Seoul, Korea

[13] Chung-Hsien Kuo, Hung-Wen Yeh and Chin-En Wu, "*Development of Autonomous Navigation Robotic Wheelchairs Using Programmable System-on-Chip Based Distributed Computing Architecture",* from IEEE Xplore, pp 2939 – 2944

[14] Chung-Hsien Kuo, Jia-Wun Siao and Kuo-Wei Chiu, "*Development of an Intelligent Power Assisted Wheelchair Using Fuzzy Control Systems*", for *IEEE International Conference on Systems, Man and Cybernetics (SMC 2008)*, pp 2578 – 2583, 2008

[15] Sehoon Oh and Yoichi Hori, "*Development of a Novel Instantaneous Speed Observer and its Application to the Power-Assisted Wheelchair Control",* from IEEE Xplore ,pp 1471 – 1476

[16] Chung-Hsien Kuo, Hung-Wen Yeh, Chin-En Wu and Ko-Ming Hsiao, "*Development of Autonomous Robotic Wheelchair Controller Using Embedded Systems"*, with reference The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON),Nov. 5-8, Taiwan, pp 3001 – 3006, 2007

[17] Rory A. Cooper, "*Intelligent Control of Power Wheelchairs"*, in Human Engineering Research laboratories for IEEE Engineering In Medicine And Biology, pp 423 -431, July/August 1995

[18] Lucas H. V. van der Woude, Arianne Bouw, Joeri van Wegen, Harry van As, Dirkjan Veeger and Sonja de Groot, "*Seat Height: Effects On Submaximal Hand Rim Wheelchair Performance During Spinal Cord Injury Rehabilitation"*, taken from Journal Compilation © 2009 Foundation of Rehabilitation Information, ISSN 1650-1977, *J Rehabil Med 41,*pp 143–149, 2009

[19] MPLAB IDE, Simulator, Editor User's Guide

[20] Martin Bates, "Interfacing PIC Microcontrollers: Embedded Design by Interactive Simulation", Publish by Elsevier, Oxford UK, 2006

[21] Proteus Design Suite 7 Release Notes

[22] MICROCIP PIC16F87XA Data Sheet 28/40/44-Pin Enhanced Flash Microcontroller

APPENDIX A

Program in Wheelchair Navigation Model

```
//=====================================================================================
//Author                  : AFIQAH YAACOB
//Project                 : Electric powered wheelchair model
//Project description     : This project is done to show the navigation based on joystick. It also equip with
                             Maxbotic range sonar sensor which function to detect hole along the road.
                            When a hole is detect, the wheelchair will reverse with high speed and buzzer
                            will turn on
// Version                : Final Year Project 2010
//=====================================================================================

//      include
//=====================================================================================
#include <pic.h>

//      configuration
//=====================================================================================
__CONFIG (0x3F32);

//      define
//=====================================================================================
#define sw1             RB0                 //to turn joystick navigation ON
#define led_blue        RD7
#define led_yellow      RD6
#define led_red         RD5
#define buzz            RB2
#define sensor          RA2                 //analog input in PIC

#define motorRA                 RC0
#define motorRB                 RC3
#define motorLA                 RD0
#define motorLB                 RD1

#define pwmL            CCPR1L              // to define the left motor speed
#define pwmR            CCPR2L              // to define the right motor speed

#define _XTAL_FREQ      20000000

//      global variable
//=====================================================================================
unsigned int result;
```

```
//        function prototype
//===============================================================================
void init(void);

void delay_ms(unsigned long data);
void delay_s(unsigned long data);

void read_adc(unsigned char channel);

void forward(void);
void backward(void);
void right(void);
void left(void);
void stop(void);

void ultrasonic(void);

void buzzer(void);

//        main function
//===============================================================================
void main(void)
{
        unsigned int FB_analog, RL_analog;

        init();

        led_blue = 1;
        led_yellow = 1;
        led_red = 1;

        while(sw1);
        buzzer();
        led_blue = 1;
        led_yellow = 1;
        led_red = 0;

        while(1)
        {
                ultrasonic();

                read_adc(0);
                RL_analog = result;
                read_adc(1);
                FB_analog = result;

                if((FB_analog>800)&&(RL_analog<800)&&(RL_analog>200))
```

```
		{
			led_blue = 0;
			led_yellow = 0;
			led_red = 1;
			pwmL = 100;
			pwmR = 125;
			forward();
		}
		else if((FB_analog<100)&&(RL_analog<900)&&(RL_analog>100))
		{
			led_blue = 1;
			led_yellow = 1;
			led_red = 0;
			pwmL = 125;
			pwmR = 100;
			backward();
		}
		else if((FB_analog<900)&&(RL_analog>900))
		{
			led_blue = 1;
			led_yellow = 0;
			led_red = 1;
			pwmL = 100;
			pwmR = 100;
			left();
		}
		else if((FB_analog<900)&&(RL_analog<100))
		{
			led_blue = 1;
			led_yellow = 0;
			led_red = 1;
			pwmL = 100;
			pwmR = 100;
			right();
		}
		else
		{
			led_blue = 1;
			led_yellow = 1;
			led_red = 0;
			stop();
		}
	}
}

//	Initailization
//	Description : Initialize the microcontroller
//==============================================================================
```

```
void init(void)
{
	//setting ADCON1 Register
	ADCON1 = 0b11000000;			// right justified, Fosc/64, set all pins analog


						// TRIS configuration 0 is output 1 is input
	TRISA = 0b111111;			// set I/O at PORDA
	TRISB = 0b00000001;			// set I/O at PORDB
	TRISC = 0b00000000;			// set I/O at PORDC
	TRISD = 0b00000000;			// set I/O at PORDD
	TRISE = 0b111;				// set I/O at PORDE

	PR2 = 255;				// period register
	T2CON = 0b00000110;			// set frequency
	CCP1CON = 0b00001100;			// set PWM mode
	CCP2CON = 0b00001100;			// set PWM mode

	buzz = 1;				//buzzer when all have been initialize
}

//	 Mode subroutine
//==============================================================================

//	Description : Delay functions
//------------------------------------------------------------------------------------------------------------------------------
void delay_ms(unsigned long data)		//delay function, the delay time in second
{
	for (;data>0;data--)
	__delay_ms(1);
}
void delay_s(unsigned long data)		//delay function, the delay time in second
{
	int i;
	do for(i=100;i>0;i--) __delay_ms(10);
	while(--data>0);
}

//	Description : ADC functions
//------------------------------------------------------------------------------------------------------------------------------
void read_adc(unsigned char channel)
{
	ADCON0 = 0b10000001 | (channel<<3);	// Fosc/64, select analog channel
	delay_ms(1);

	GODONE = 1;			// GODONE is the bit 2 of the ADCON0 register
	while(GODONE==1);		// ADC start, GODONE = 0 after finish ADC progress
	result = ADRESH;		//all ADC result will arrange into specific address
```

```
	result = result << 8;
	result = result | ADRESL;
}

//	Description : Motor functions
//-----------------------------------------------------------------------------------------------------------------------------------
void forward(void)
{
	motorLA = 0;
	motorLB = 1;
	motorRA = 1;
	motorRB = 0;
}
void backward(void)
{
	motorLA = 1;
	motorLB = 0;
	motorRA = 0;
	motorRB = 1;
}
void right(void)
{
	motorLA = 0;
	motorLB = 1;
	motorRA = 0;
	motorRB = 0;
}
void left(void)
{
	motorLA = 0;
	motorLB = 0;
	motorRA = 1;
	motorRB = 0;
}
void stop(void)
{
	motorLA = 1;
	motorLB = 1;
	motorRA = 1;
	motorRB = 1;
}

//	Description : Ultrasonic function
//-----------------------------------------------------------------------------------------------------------------------------------
void ultrasonic(void)
{
	int distance;
	read_adc(2);
```

```
	distance = result;

	if(distance>100)			// check if distance more than 25
	{
		pwmL = 200;		// then backward with high speed and on the buzzer
		pwmR = 200;
		backward();
		buzz = 0;		// buzzer on
		delay_ms(500);
	}
	else buzz = 1;
}

//	Description : Buzzer functions
//-----------------------------------------------------------------------------------------------------------------------------------
void buzzer(void)
{
	buzz = 0;
	delay_ms(100);
	buzz = 1;
	delay_ms(100);
}
```

APPENDIX B

Schematic Circuit Diagram