

UNIVERSITI MALAYSIA PAHANG

**BORANG PENGESAHAN STATUS TESIS**

JUDUL: **DESIGN OF OBSTACLE AVOIDANCE ROBOT**

SESI PENGAJIAN: 2011/2012

Saya LEE LE XIN ( 880903-23-6021 )

mengaku membenarkan tesis (Sarjana Muda/Sarjana/Doktor Falsafah)\* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. \*\*Sila tandakan ( √ )

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:



(TANDATANGAN PENULIS)



(TANDATANGAN PENYELIA)

Alamat Tetap:

**37, JALAN SENTOSA 2,**  
**TAMAN SENTOSA,**  
**KG. TENGAH, 85000**  
**SEGAMAT, JOHOR**

**PROF. IR. DR. AHMAD FAIZAL**  
**MOHD. ZAIN, PEng, FASc, FIEM**  
**SENIOR MIEEEE**

Tarikh: 9/7/12

Tarikh: 12/7/12

- CATATAN:
- \* Potong yang tidak berkenaan.
  - \*\* Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
  - ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

# DESIGN OF OBSTACLE AVOIDANCE ROBOT

LEE LE XIN

Report submitted in partial fulfillment of the requirements  
for the award of the degree of  
Bachelor of Manufacturing Engineering

Faculty of Manufacturing Engineering  
UNIVERSITY MALAYSIA PAHANG

JUNE 2012

PERPUSTAKAAN UNIVERSITI MALAYSIA PAHANG	
No. Perolehan <b>067463</b>	No. Panggilan TJ 211 144
Tarikh <b>11 OCT 2012</b>	2012 15 Bc.

**UNIVERSITY MALAYSIA PAHANG**  
**FACULTY OF MANUFACTURING ENGINEERING**

I certify that the project entitled "Design of Obstacle Avoidance Robot" is written by Lee Le Xin. I have examine the final copy of this project in my opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering. I herewith recommend that it be accepted in partial fulfillment of the requirement for the degree of Bachelor of Manufacturing Engineering.

Examiner : *Nafizuan Mat Yahya*

Signature:



**NAFRIZUAN BIN MAT YAHYA**  
**HEAD OF PROGRAMME (MECHATRONIC)**  
**FACULTY OF MANUFACTURING ENGINEERING**  
**UNIVERSITI MALAYSIA PAHANG**  
**26600 PEKAN**  
**PAHANG DARUL MAKMUR**  
**TEL: 09-424 5836 FAX: 09-424 5888**

**SUPERVISOR'S DECLARATION**

"I hereby declare that I have read this thesis and in my opinion this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Manufacturing Engineering"

Signature



Name of Supervisor : Prof. Ir. Dr. Ahmad Faizal Mohd. Zain, PEng., FASc, FIEM,  
Senior MIEEE

Position

:

Date


:

**PROF. IR. DR. AHMAD FAIZAL BIN MOHD ZAIN**  
FAKULTI KEJURUTERAAN PEMBUATAN  
UNIVERSITI MALAYSIA PAHANG  
26600 PEKAN  
PAHANG DARUL MAKMUR  
TEL: 09-424 5801 FAXS: 09-424 5888

12/7/12

**STUDENT'S DECLARATION**

**I hereby declare** that the work in the project is my own except for quotation and summaries which have been duly acknowledged. The project has not been accepted for any degree and is not concurrently submitted for award of other degree.

**Signature** : 

**Name** : LEE LE XIN

**Id Number** : FA08006

**Date** : 9/7/12

## DEDICATION

Specially dedicated to  
My beloved parents, brother, sister  
and all of my friends.

## ACKNOWLEDGEMENT

**I am** greatly indebted to my supervisor, Prof. Ir. Dr. Ahmad Faizal Mohd. Zain for his **advice** and guidance throughout my project.

**I would** like to thank my family member for supporting me and their caring throughout **my** four years study in University Malaysia Pahang.

**Special** thanks to FKP staffs helping and giving suggestion for me to complete my **project**. And also thanks to my friend who always help me to suggest and giving me the **criticisms**. Thank you all.

**Finally**, I would like to express my thanks to those who involves directly or indirectly in **completion** of my project.

## **ABSTRACT**

Recently, there are many types of robots available in market, either in form of pets, toys, cleaners and etc. Robots are utilized in many applications to assist human-being. The purpose of this project is to develop an autonomous robot that can move by itself without continuous human guidance. This robot will be controlled using the SK40C controller board that used an infrared sensor, microcontroller PIC 18F4550 and also a dc geared motor and a 16x2 LCD screen. When the robot is on, the LCD screen will show "Avoidance robot ON" and the buzzer will sound a beep continued by a move forward. When it senses an obstacle in front, it will reverse and then either turn left or right depending on the left sensor or right sensor that detect the obstacle then it will continue to move forward. When it senses an obstacle at the left, the LCD screen will show "Obstacle at front, reversing", then followed by "Turn right, move forward". As a result, this robot is useful as a baby walker to avoid the baby walker from hitting thing that may cause harm to the baby.



## ABSTRAK

Sejak kebelakangan ini, terdapat banyak jenis robot yang boleh didapati di pasaran, sama ada dalam bentuk haiwan peliharaan, mainan, pembersih, robot dan sebagainya telah digunakan dalam pelbagai kegunaan untuk membantu manusia. Tujuan projek ini adalah untuk membina sebuah robot yang boleh bergerak sendiri tanpa bimbingan manusia yang berterusan. SK40C akan digunakan dalam robot ini, penderia inframerah, microcontroller PIC18F4550, dc motor dan juga skrin LCD akan digunakan. Apabila robot dinyalakan "Autonomous Robot ON" akan ditunjukkan pada skrin LCD dan bunyi beep akan keluar dan juga robot akan mula gerak ke hadapan. Apabila halangan di kesan dihadapan, ia akan berundur kemudian sama ada belok kiri atau ke kanan bergantung sensor mana yang mengesan halangan di depan. Apabila halangan pada sebelah kiri, LCD screen akan menunjukkan "Halangan di kiri, berundur", kemudian diikuti dengan "belok kanan" dan bergerak ke hadapan. Robot ini adalah berguna jika digunakan pada kerusi roda bayi, ia boleh menghalang bayi daripada mengelak pada halangan yang mungkin membahayakan keselamatan bayi.

## TABLE OF CONTENTS

<b>TITLE PAGE</b>	i
<b>EXAMINER APPROVAL</b>	ii
<b>SUPERVISOR'S DECLARATION</b>	iii
<b>STUDENT'S DECLARATION</b>	iv
<b>DEDICATION</b>	v
<b>ACKNOWLEDGEMENT</b>	vi
<b>ABSTRACT</b>	vii
<b>ABSTRAK</b>	viii
<b>TABLE OF CONTENTS</b>	ix
<b>LIST OF TABLES</b>	xii
<b>LIST OF FIGURES</b>	xiii
<b>LIST OF ABBREVIATIONS</b>	xv
<b>LIST OF FIGURE</b>	xvi
<b>LIST OF EQUATION</b>	xvii
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Background	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scope of Project	4
1.5 Thesis Overview	5
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
2.0 Introduction	6
2.1 Types of Robot	6
2.2 Robotic	8
2.3 PIC Controller	10
2.3.1 Microcontroller	10
2.3.2 Types of Microcontroller	11

2.4 Control Theory	13
2.5 Infra –Red Sensor	14
2.6 Dc-Geared Motor	15
<b>CHAPTER 3 HARDWARE &amp; SOFTWARE DESIGN</b>	<b>16</b>
3.0 Introduction	16
3.1 Hardware Review	18
3.1.1 Circuit Overview	18
3.1.2 PIC Microcontroller Start-up Kit (SK40C)	20
3.1.3 PIC 18F4550 Microcontroller	21
3.1.4 Infrared Sensor	22
3.1.5 L293D Driver	24
3.1.6 2X16 Liquid Crystal Display	26
3.1.7 Piezo Electric Buzzer	27
3.1.8 UIC00B USB ICSP PIC Programmer	28
3.2 Mechanical Design	29
3.2.1 DC Geared Motor	29
3.2.2 Tires	30
3.2.3 Chassis of Robot	31
3.3 Software Design	31
3.3.1 Background	31
3.3.2 Algorithm	32
3.3.3 PCW IDE Compiler	33
3.3.4 PICKIT 2 Programmer	35
<b>CHAPTER 4 RESULT &amp; DISCUSSION</b>	<b>38</b>
4.1 DC Motor Testing	38
4.1.1 Robot Moving Forward	39
4.1.2 Robot Reverse	39
4.1.3 Robot Turn Left	40
4.1.4 Robot Turn Right	42
4.2 IR Emitter and Receiver Testing	45
4.3 Equation Forming Using ADC	48
4.4 Limitation of the robot	53

<b>CHAPTER 5 CONCLUSION</b>	<b>54</b>
5.1 Conclusion	54
5.1 Future Development	55
<b>REFERENCES</b>	<b>56</b>
<b>APPENDIX</b>	<b>58</b>

**LIST OF TABLES**

<b>Table No.</b>	<b>Title</b>	<b>Page</b>
2.0	List of PIC controller type	12
4.1	Logic and voltage values of L293D when robot moving forward	39
4.2	Logic and voltage values of L293D when robot moving reverses	40
4.3	Logic and voltage values of L293D when robot moving left	41
4.4	Logic and voltage values of L293D when robot moving right	43
4.5	Distance and voltage value for left IR sensor	46
4.6	Distance and voltage value for right IR sensor	46
4.7	Distance and decimal value for left IR sensor before(1kohm) and after (30kohm) preset value adjust	48
4.8	Distance and decimal value for right IR sensor before(1kohm) and after (30kohm) preset value adjust	51

4.5	Obstacle at distance 9cm	45
4.6	Obstacle at distance 2cm	45
4.7	Graph of Voltage versus Distance for left IR Sensor	46
4.8	Graph of Voltage versus Distance for right IR Sensor	47
4.9	Get Digital value from IR sensor using ADC	48
4.10	Sample of decimal value for left sensor	48
4.11	Sample of decimal value for right sensor	49
4.12	Decimal versus Distance left IR sensor before adjust preset	49
4.13	Decimal versus Distance left IR sensor after adjust preset	50
4.14	Decimal versus Distance right IR sensor before adjust preset	51
4.15	Decimal versus Distance right IR sensor after adjust preset	52
5.1	Complete Autonomous Robot	52

**LIST OF ABBREVIATIONS**

<b>PIC</b>	<b>Programmable interface Controller</b>
<b>IC</b>	<b>Integrated circuit</b>
<b>V</b>	<b>Volt</b>
<b>dc</b>	<b>Direct current</b>
<b>I/O</b>	<b>Input/Output</b>
<b>A</b>	<b>Ampere</b>
<b>IR</b>	<b>Infrared</b>
<b>cm</b>	<b>Centimeter</b>
<b>ADC</b>	<b>Analog To Digital converter</b>
<b>LCD</b>	<b>Liquid Crystal Display</b>
<b>PM</b>	<b>Permanent Magnet</b>
<b>DC</b>	<b>Direct Current</b>
<b>RPM</b>	<b>Revolutions per minute</b>

**LIST OF APPENDIX**

<b>Appendix No.</b>	<b>Title</b>	<b>Page</b>
<b>A</b>	Autonomous Robot Programming	55
<b>B</b>	Data Sheet	63



## LIST OF EQUATION

Equation No.	Title	Page
4.1	$Y = mX + C$	50
4.2	$Y = -7.2X + 126$	50
4.3	$Y = mX + C$	52
4.4	$Y = -5X + 100$	52

# CHAPTER1

## INTRODUCTION

### 1.1 BACKGROUND

The term “robot” was introduced by Czech dramatists Joseph Capek in his brother Karel Capek play Rossum’s Universal Robots in 1921. The word robot comes from the Czech word “robota” meaning “labor”. Later, in 1942, Isaac Asimov built upon Capek’s ideas and defined the term “robotics” and outlined the three rules of robotics in his books that is may not injure must obey the orders must obey the orders. In Europe ,robotics is defined as the science of robotology and the robotology is defined as the meaning of putting the robot machines together and to do work. [1,2]

Autonomous robots are robots that can perform desired tasks in unstructured environments without continuous human guidance. Many kinds of robots have some degree of autonomy. Different robots can be autonomous in different ways. A high degree of autonomy is particularly desirable in fields such as space exploration, cleaning floors, mowing lawns, and waste water treatment.

For humans, the ability to navigate our thinking movement is controlling using our brains. But for mobile robotic systems, navigation in dynamic real-world environments is an extremely complex and challenging task. Complex structure and the movement of robot is more harder depend to its complexity of design, mostly the design of movement of robot is copying from human motion activity and some are copy from animals or insects such as spider.

Autonomous robot has the ability to gain information about the environment, ~~work~~ for an extended period without human intervention, move either all or part of itself ~~throughout~~ its operating environment without human assistance, and can avoid ~~situations~~ that are harmful to people, property, or itself.

Before the autonomous robot was invented, there were only ordinary robots. These robots were depending on human control. These robots did not have any self ~~avoiding~~ system toward obstacles; all is totally control by human. Therefore to over ~~comes~~ those problems, the autonomous robot were invented to replace human in ~~doing~~ dirty, difficult and *dangerous* work such as denoting the bombs.

Therefore, this project is to develop an autonomous robot that can move to the ~~destination~~ without getting block by obstacles and can show the information to LCD ~~screen~~. The basic concept of this project is to design a robot that can move forward and ~~backward~~ without human guidance. The PIC controller will be used as the "brain" of the ~~robot~~ to control its system. The infrared sensor will be using as the guidance to track the ~~obstacles~~ that block the path.

## 1.2 PROBLEM STATEMENT

An important problem that a navigating robot is facing in a crowded ~~environment~~ is that the robot can be easily get blocked by moving humans and ~~obstacles~~. In such cases, the robot becomes immobilized and not able to continue its ~~movement~~ towards its goal position until the moving obstacles and humans free from its ~~way~~. To avoid ever getting in such a situation, many researchers have tried to predict ~~the~~ motion of humans and obstacles. The future motion prediction of humans and ~~obstacles~~ allows the robot to estimate if the way it follows is going to be blocked and ~~this~~ change direction before it ever faces this situation.

Future motion prediction is an intrinsic behavior of humans. Consider the ~~example~~ of a man trying to cross a street. A man looks at his left and right to see if any ~~vehicles~~ are approaching. If there are no vehicles he is certain that he can cross the ~~street~~ safely and so he does. In the case that a vehicle is approaching, he tries to estimate

~~how~~ long it will take for the vehicle to reach the point that he stands and then decides if ~~he~~ should cross the street. This is in effect a predictive behavior. It would be desirable ~~for~~ an autonomous robot to develop a similar behavior.

### **L3 OBJECTIVE**

~~The~~ objective of this project is to:

**1)** To develop an obstacle robot that can move it-self without continuous human ~~guidance~~ by using PIC controller as the “brain” of robot to operate.

**PIC18F8550** will be use as controller in this project. So by programming the controller ~~the~~ movement of the robot can be predicted using programming.

**2)** To perform detection of obstacles using Infra-red sensors and will be giving respond ~~from~~ PIC controller to change robot path.

**Infrared** sensor will be installed on the robot .By using the sensor when the robot is near ~~to~~ the obstacle, the signal will be sent to microcontroller to let the controller decide the ~~step/process~~ should be taken ,to change to which direction.L293D motor driver will be ~~use~~ to control the dc motor rotation direction.

**3)** To produce a robot that wills use PIC as the system to generate the data receive from ~~analog~~ output of infra-red sensors become digital and show the result or output using LCD screen.

~~The~~ analog output ~~from~~ infrared sensor will be connected to the build in ADC inside the **PIC18F8550** and 2 pin of RA pin will be define as channel of ADC input.

## **1.1 SCOPE OF PROJECT**

~~These~~ scopes are proposed for this project:

1) The robot will move forward at show "Avoidance Robot ON" on LCD screen at ~~starting~~ and the buzzer beep to show the robot is on.

~~When~~ the robot is switch on, the message "Avoidance Robot ON" will be display on LCD screen and both left and right motor will rotate forward to let the robot move forward.

2) When the obstacle is being senses at left, it will show "Obstacle at left,reverse". Then follow by "Turn right"

To let the motor reverse both motor needs to rotate backward, for the robot to turn right, ~~the left~~ motor will be rotating forward while right motor will be rotating backward.

3) When the obstacle is being senses at right, it will show "Obstacle at right,reverse" Then follow by "Turn left"

To let the motor reverse both motor needs to rotate backward for the robot to turn right, ~~the left~~ motor will be rotating backward while right motor will be rotating forward.

## **LS THESIS OVERVIEW**

**This thesis is a combination of 5 chapters that contains the Introduction, Literature Review, Hardware & Software design, Result & Discussion and last is Conclusion.**

**Chapter 1 is the introduction of the project. The background, objective and the scope of project are explained in this chapter.**

**Chapter 2 is the literature review that made from several journal that been refer and also consist of the methodologies to done the project.**

**Chapter 3 is the explanation of hardware and software design of the project. In hardware design, the electronics part and mechanical parts will be explain, the electronic part will be focusing of controller board, IR sensor, L293D motor driver and also LCD screen while mechanical parts will explain about the construction of the robot. For software design, the programming of the PIC18F4550 will be explained.**

**Chapter 4 will show all the results and the analysis of the project. All of the result obtain will be analyzed and the comment will be given due to the result getting.**

**For Chapter 5 the conclusion and discussion will be explain in this chapter.**

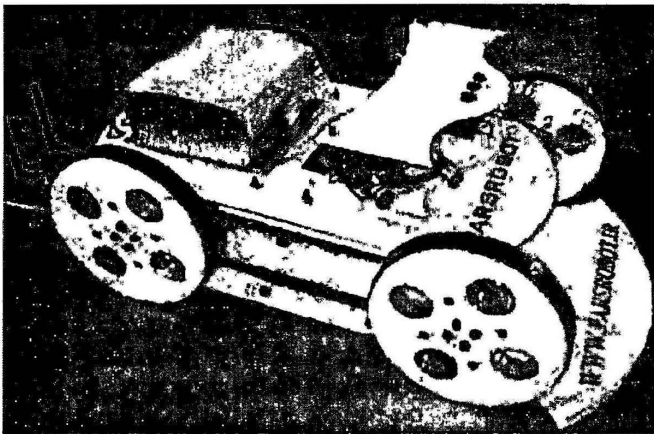
## CHAPTER 2

### LITERATURE REVIEW

#### 2.0 INTRODUCTION

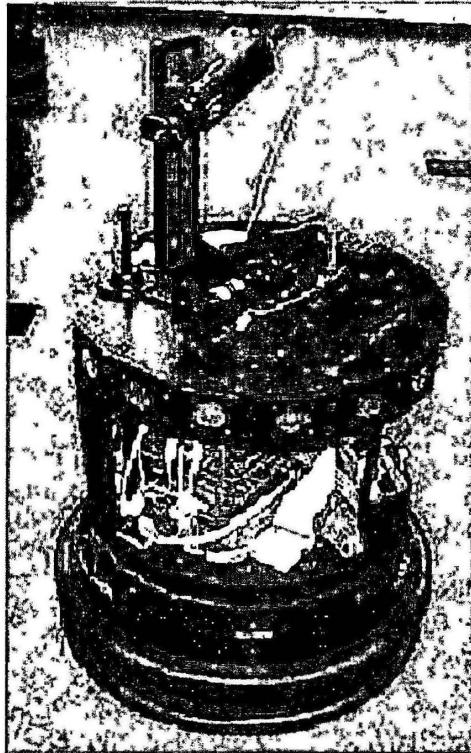
Conducting literature review prior to begin a research project is vital as this will supply the researcher with much needed additional and information on the methodologies and technology available used by other research counterparts around the world on the topic. This chapter provides a condensed summary of literature reviews on key topics related to types of robot.

#### 2.1 TYPES OF ROBOT



**Figure 2.1:** Autonomous Minesweeping Robot (Aria)

**Aria Robot** was a minesweeping robot that designed to use for reduction of the ~~mine~~ activities in dangerous environment. Mine detection is a risky action that can be ~~done~~ by either human or robot. Aria Robot is an autonomous four wheels robot that is ~~designed~~ to search through an arbitrary minefield for detecting landmines. Wireless ~~connection~~ is used to established between the local controller on the robot and general ~~controller~~ that is installed at the outside of the minefield to handle the whole navigation ~~procedure~~. A personal computer (PC) acts as the general controller which is responsible ~~for~~ ~~mapping~~ the landmines and determining some general settings for local controller. ~~The~~ local controller navigates the robot based on the feedbacks from several sensors to ~~detect~~ the barriers and landmines such as digital compass, optical shaft encoder, current ~~sensor~~, ultrasonic range finder.[5]



**Figure 2.2:** The Nomad 200 mobile Robot (Nomadic Technologies, Inc.)



The Nomad 200 has three wheels controlled by two motors. One motor controls the synchronous translation of the three wheels. The other motor enables the three wheels to rotate together. The robot has a turret which, under the control of a third motor, can rotate independently from the base. The robot has a zero turning-radius, it can rotate around its center. Both the steering and drive motors have encoders which enable the measurement and control of both the Cartesian location of the robot, The robot has 16 sonar range sensors, and 16 infrared range sensors. Both types of sensors are equally spaced around the turret. The sonar system uses standard Polaroid transducers, that are based on the usual technique of measuring the time of-flight of an acoustic wave, from emission to reception after being reflected by a detected object. Each sonar sensor is able to measure distances from 15.2 cm to 10.6 m. Each infrared range sensor is composed of one photodiode receiver, placed between two emitters, with all the three elements horizontally disposed. The infrared sensors are used to measure distances to objects less than 60 cm away.[3][4]

## 2.2 ROBOTIC

Robotics is a revolution of industrial technology. The term of robotics means a different perception and understanding to different people. Some people imagined that robot as Bumble-B and Optimus Prime from the Transformer movies or others movies characters as real robot should be. But in the reality, robot can be formed in many shape and type as long as it does fulfill the characteristics of robot. The evolution of robotics started in the twenty first century. The 'robot' term was firstly coined in early 1920 by a Czechoslovakian dramatist, Karel Capek in his play entitles "Rossum's universal robots". *Robota* is the original term of robot that widely used since it has been introduced to the public. *Robota* is a czech word which means 'salve laborer'. On 1942 science fiction author Isaac Asimov used the term 'robotics' in his short story "Runaround" [1][2].

Traditionally, robots are running in carefully structured environments such as factories, where, while difficult, the designing process of their morphologies and tasks, have been manageable. In the near futures, robots are required to run in highly dynamic and unpredictable environments like in hospitals, homes, schools and streets, often

without human supervisions. Robots increasingly important roles in our daily life, the increase in complexity of the tasks and sometimes physical forms or morphologies of the artifacts consequently requires complex assembling and controlling procedures of them, which soon will be unmanageable by the traditional manufacturing process. [6]

Three Laws of robotics have been introduced in "Runaround" described the three basic of rules that robot should follow to operate without harming or cause injury to human. The laws are [7][8]:

- a) Robot must not injure human
- b) A robot must always follow the direction or order given to it by the human being as much as it has not contrast or conflict to first law.
- c) A robot must protect its own existence as long as it does not make conflict to first and second law.

The word of robot widely used to describe some electro-mechanical device that is given a set of instruction, which may be partially or fully automated. Robot is defined as a machine that can be programmed in order to accomplish any task provided. Robot Institute of America defines a robot as:

A re-programmable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for performance of a variety of tasks [6].

Besides the conventional use of industrial robots in industry and the use of fully autonomous robots for collecting samples on other planets (Bares et al. 1989) a new application area for robots has come into being. In these new application fields, humans are the most important factor, who use the robots as partly autonomous or as simply controllable tools. [6]

## 2.3 PIC CONTROLLER

### 2.3.1 Microcontroller

Microcontrollers contain at least two primary components – random access memory (RAM), and an instruction set. RAM is a type of internal logic unit that stores information temporarily. RAM contents disappear when the power is turned off. While RAM is used to hold any kind of data, some RAM is specialized, referred to as registers. The instruction set is a list of all commands and their corresponding functions. During operation, the microcontroller will step through a program (the firmware). Each valid instruction set and the matching internal hardware that differentiate one microcontroller from another. [9]

Most microcontrollers also contain read-only memory (ROM), programmable read-only memory (PROM), or erasable programmable read-only memory (EPROM). All of these memories are permanent: they retain what is programmed into them even during loss of power. They are used to store the firmware that tells the microcontroller how to operate. They are also used to store permanent lookup tables. Often these memories do not reside in the microcontroller; instead, they are contained in external ICs, and the instructions are fetched as the microcontroller runs. This enables quick and low-cost updates to the firmware by replacing the ROM.

The number of I/O pins per controllers varies greatly, each I/O pin can be programmed as an input or output (or even switch during the running of a program). The load (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. [9][10]

### 2.3.2 Types of Microcontroller

Microcontrollers PICmicro MCU from Microchip Company divided into 4 large families. Each family has a variety of components that provide built-in special features:

1. **The first family, PIC10 (10FXXX) - is called Low End.**

The PIC10FXXX devices from Microchip Technology are low-cost, high-performance, 8-bit, fully static, Flash-based CMOS microcontrollers. They employ a RISC architecture with only 33 single-word/ single-cycle instructions. The 12-bit wide instructions are highly symmetrical. The easy-to-use and easy to remember instruction set reduces development time significantly. The PIC10FXXX devices contain an 8-bit ALU and working register.

2. **The second family, PIC12 (PIC12FXXX)– is called Mid-Range.**

The PIC12FXXX most popular among these starter their way in this field. Mid-Range devices feature 14-bit program word architecture and are available in 8 to 64-pin packages that offer an operating voltage range of 1.8-5.5V, small package footprints, interrupt handling, an 8-level hardware stack, multiple A/D channels and EEPROM data memory. Mid-range devices offer a wide range of package options and a wide range of peripheral integration. These devices feature various serial analog and digital peripherals, such as: SPI, I2C™, USART, LCD and A/D converters.

3. **The third family is PIC16(16FXXX)**

With six variants ranging from 3.5K-14 Kbytes of Flash memory, up to 256 bytes of RAM and a mix of peripherals including EUSART, CCP and onboard analog comparators. These devices are well suited for designers with applications that need more code space or I/O than 14-pin variants supply, and are looking to increase system performance and code efficiency by employing hardware motor control and communications capability.

#### 4. The fourth family is PIC 17/18(18FXXX).

The PIC18 family utilizes a 16-bit program word architecture and incorporates an advanced RISC architecture with 32 level-deep stack, 8x8 hardware multiplier, and multiple internal and external interrupts. With the highest performance in Microchip's 8-bit portfolio, the PIC18 family provides up to 16 MIPS and linear memory. PIC18 is the most popular architecture for new 8-bit designs where customers want to program in C language.

**Table 2.0 :** List of the PIC controller type

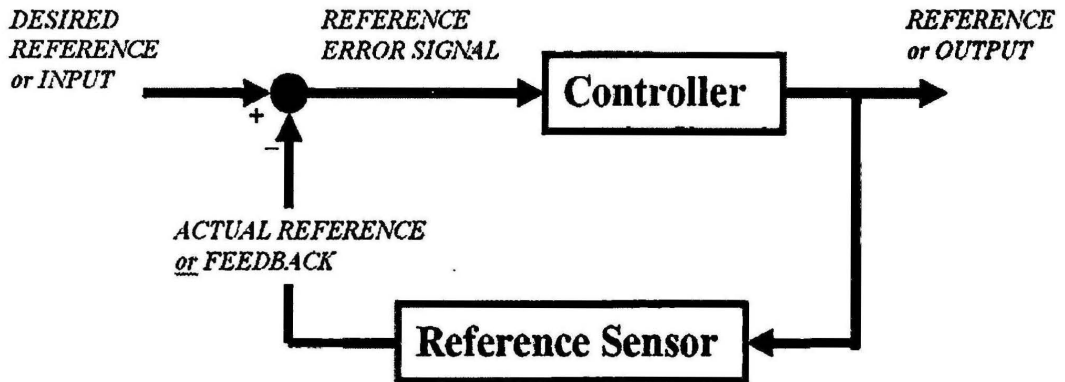
PIC #	# of pins	I/O pins	Program ROM words	File RAM bytes	EEPROM bytes	Analogue inputs	Timers 8/16 bits	Max clock (MHz)	Internal osc. (MHz)	In-circuit debug	Serial comms
12F675	8	6	1k	64	128	4x10-bit	1/1	20	4	YES	NO
16F628A	18	16	2k	224	128	NO	2/1	20	4	NO	UART
16F630	14	12	1k	64	128	NO	1/1	20	4	YES	NO
16F648A	18	16	4k	256	256	NO	2/1	20	4	NO	UART
16F676	14	12	1k	64	128	8x10-bit	1/1	20	4	YES	UART
16F73	28	22	4k	192	NO	5x8-bit	2/1	20	NO	NO	ALL
16F77	40	33	8k	368	NO	8x8-bit	2/1	20	NO	NO	ALL
16F818	18	16	1k	128	128	5x10-bit	2/1	20	8	YES	I2C,SPI
16F84	18	13	1k	64	64	NO	1/0	10	NO	NO	NO
16F84A	18	13	1k	64	64	NO	1/0	20	NO	NO	NO
16F88	18	16	4k	368	256	7x10-bit	2/1	20	8	YES	ALL
16F874A	40	33	4k	192	128	8x10-bit	2/1	20	NO	YES	ALL
16F876A	28	22	8k	256	368	5x10-bit	2/1	20	NO	YES	ALL
16F877A	40	33	8k	256	368	8x10-bit	2/1	20	NO	YES	ALL
18F2320	28	25	4k	512	256	10x10-bit	1/3	40	8	YES	ALL
18F6520	64	52	16k	2048	1024	12x10-bit	1/3	40	NO	YES	ALL
18F8621	80	68	32k	3840	1024	16x8-bit	1/3	40	10	YES	I2C,SPI
18F8720	80	68	64k	3840	1024	16x10-bit	1/3	40	NO	YES	ALL

## 2.4 CONTROL THEORY

Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems. The desired output of a system is called the reference. When one or more output variables of a system need to follow a certain reference over time, a controller manipulates the inputs to a system to obtain the desired effect on the output of the system [11].

If we consider an automobile cruise control, it is design to maintain the speed of the vehicle at a constant speed set by the driver. In this case the system is the vehicle. The vehicle speed is the output and the control is the vehicle throttle which influences the engine torque output. One way to implement cruise control is by locking the throttle at the desired speed but when encounter a hill the vehicle will slow down going up and accelerate going down. In fact, any parameter different than what was assumed at design time will translate into a proportional error in the output velocity, including exact mass of the vehicle, wind resistance, and tire pressure. This type of controller is called an open-loop controller because there is no direct connection between the output of the system (the engine torque) and the actual conditions encountered mean the system does not and cannot compensate for unexpected forces [11].

For a closed-loop control system, a sensor will monitor the vehicle speed and feedback the data to its computer and continuously adjusting its control input or the throttle as needed to ensure the control error to a minimum therefore maintaining the desired speed of the vehicle. Feedback on how the system is actually performing allows the controller (vehicle's on board computer) to dynamically compensate for disturbances to the system, such as changes in slope of the ground or wind speed [11]. An ideal feedback control system cancels out all errors, effectively mitigating the effects of any forces that may or may not arise during operation and producing a response in the system that perfectly matches the user's wishes [11]



**Figure 2.3:** Concept of the Feedback Loop to Control the Dynamic Behavior

## 2.5 INFRA-RED SENSORS

Infrared sensor is best suited for the robot. The main reason for using infrared sensor is it enables the robot to efficiently avoid the robot at a range before the robot crashes them. Infrared sensor is also low cost compared to the other sensors. The drawback for infrared sensor is the less effective to sense when the obstacle is a dark and soft object. Each infrared sensor consist an infrared emitter and an infrared receiver. The detection range must not be set to far because the robot will be too sensitive to the environment.

The complete distance information between the robot and obstacles, which is obtained by using infrared sensors and the binocular sensor can real-time help the robot perceiving its location relative to all kinds of obstacles under the unknown environment.[12]

## 2.6) DC-GEARED MOTOR

DC gear motors are widely used in various modern transmission systems. In general, they are the combination of DC motors and gear reducer. Gear reduction is needed when the output speed of a motor is too fast for normal use. The output of a gear motor is provided by a gear reducer that is coupled with a DC motor as the input. The permanent magnet (PM) converts electrical energy into mechanical energy through interaction of two magnetic fields. One is by electrical current flowing in the motor windings. The field of electromagnet will result in a torque that rotates the rotor.

### Advantages of DC-g geared motor

- I. East to understand the design
- II. Easy to control the speed
- III. Providing a high rate of power at range of different speeds

### Disadvantages of DC-g geared motor

- I. lower speed
- II. mechanical wear
- III. Consequent need for servicing.

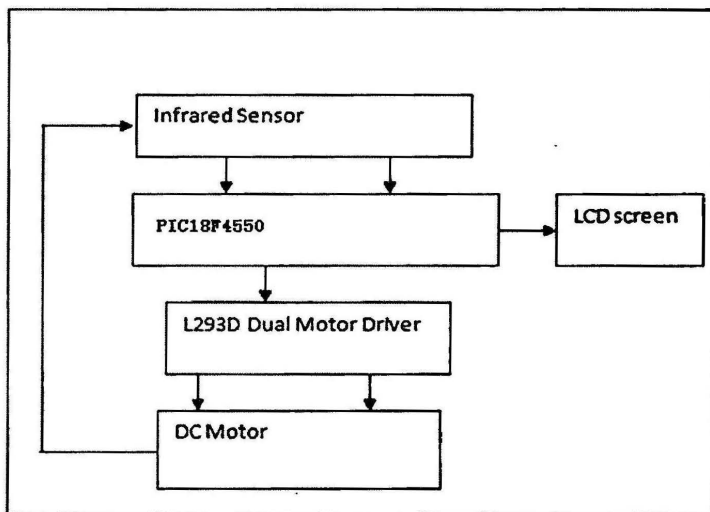


## CHAPTER3

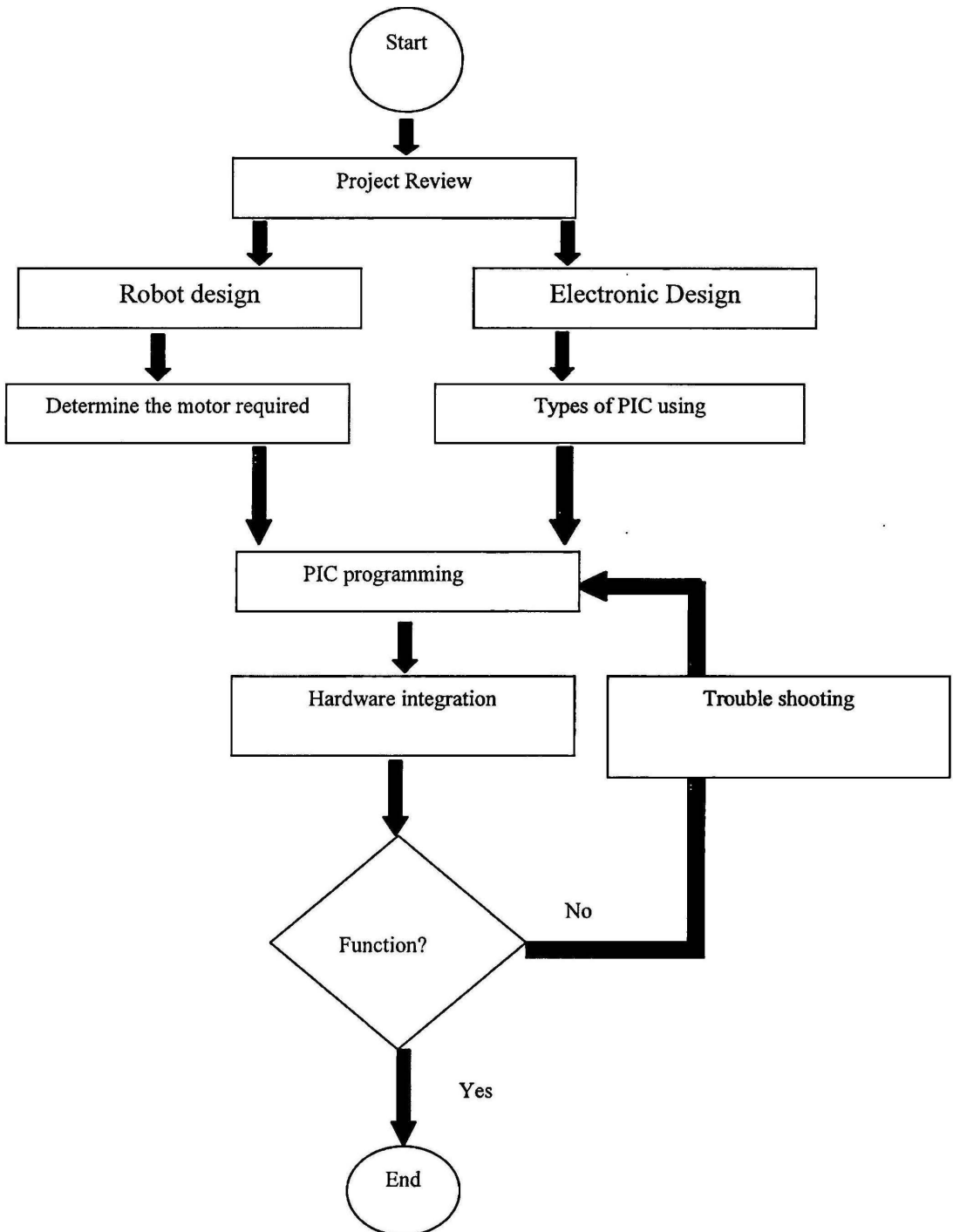
### HARDWARE & SOFTWARE DESIGN

#### 3.0 INTRODUCTION

To design an obstacle avoidance robot, by using the knowledge learn from the Control system engineering and operate with PIC controller. PIC controller will be the main system to operate the robot, by getting the feedback from infrared sensor and the range of robot travel will be receive by microcontroller PIC18F4550 to perform the reaction when meet the obstacle.L293D will be controlled by program that write into the microcontrollers. Figure 3.1 show the block diagram of the robot.



**Figure 3.1:** Block diagram of the hardware



**Figure 3.2:** Flow chart software controller

### 3.1 HARDWARE REVIEW

Generally, six important hardware components used in this project, which are two dc geared motor and motor driver, microcontroller, IR sensor buzzer and also LCD screen.

#### 3.1.1 Circuit Overview

The autonomous robot schematic circuit diagram is show in Figure 3.3. Port D0,D1,D2,D3,D4,D5,D6,D7(pin19,20,21,22,27,28,29,30) of PIC18F4550 is connected to LCD as data input/output line, while B4 and B5(pin 37and38) is function as instruction/data register selection and write selection over LCD. Port B0,B1,B,B3 (pin33,34,35,36) are connected to motor driver ST L293D. O1 and O2 of motor are connected to the right motor while O3 and O4 are connected to left motor. Enable1 (pin1)and Enable2(pin16) of L293D are connected to +5V to enable the driver. Vs(pin8) and Vss (pin16)of L293D are connected to the +9V batteries while pin4,5,12,13 are grounded. For B6(pin39)is connected to buzzer .RE0 and RE1 (pin8and9) is connected to output of IR sensor as ADC. The Port A is unused in the circuit, for those unused ports can be used for future development.

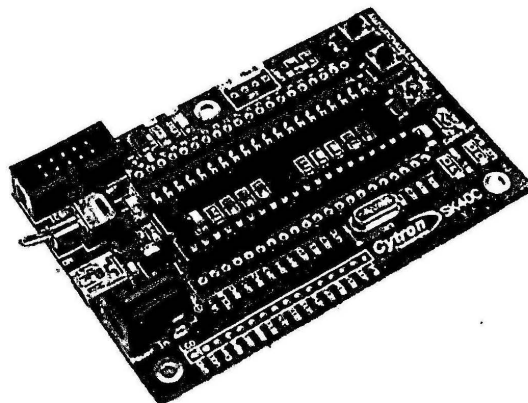


### 3.1.2 PIC Microcontroller Start-up Kit(SK40C)

This circuit board is designed to offer an easy to start board for PIC MCU user. All interface and program should be developed by user. This board comes with basic element for user to begin project development. It offer plug and use features 33 I/O pins which are nicely labeled to avoid miss-connection by users.. There is also provided connector for UIC00A (low cost USB ICSP PIC Programmer) for simple and fast method to load program.

SK40C come with:

- 5V voltage regulator (1A max)
- 20MHz crystal oscillator
- Reset button
- 1 x programmable push button
- Connector for UIC00A
- On/Off switch for main power
- DC adaptor socket as power input
- And all the necessities to eliminate users difficulty in using PIC
- UART communication
- USB on board.



**Figure 3.4 :** PIC Microcontroller Start-up Kit (SK40C)

### 3.1.3 PIC18F4550 Microcontroller

The PIC18 microcontroller is often used to function as the brain of the robot. The PIC18 is an 8-bit programmable microcontroller that has many unique onboard features and peripherals. The PIC18F4550 can be programmed in either assembly or C. The PIC18 microcontroller was programmed in C language and serve as the digital controller to control the robot. The specific PIC18 model that was used for this project is the PIC18F4550. Some of the features and peripherals onboard the PIC18F4550 include;

- Memory Endurance
- Self-Programmability
- Enhanced CCP Module
- 10-bit A/D Converter
- Enhanced Flash Program Memory Type
- 32768 bytes Program Memory Size
- 2048 bytes RAM
- 256bytes Data EEPROM
- 34 I/O pins

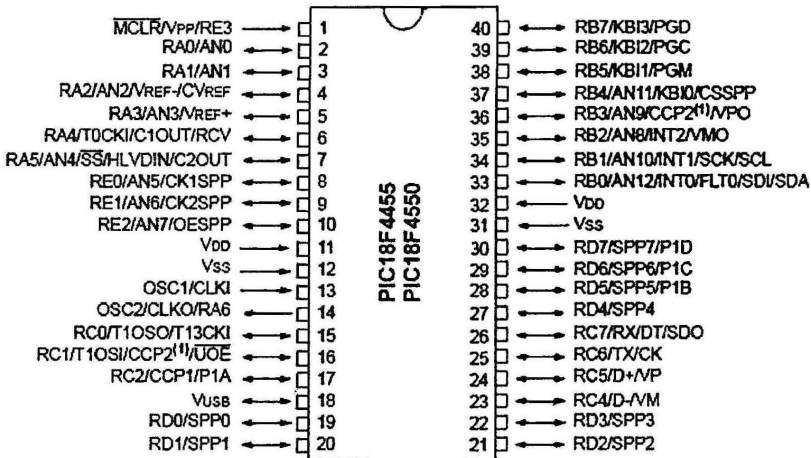
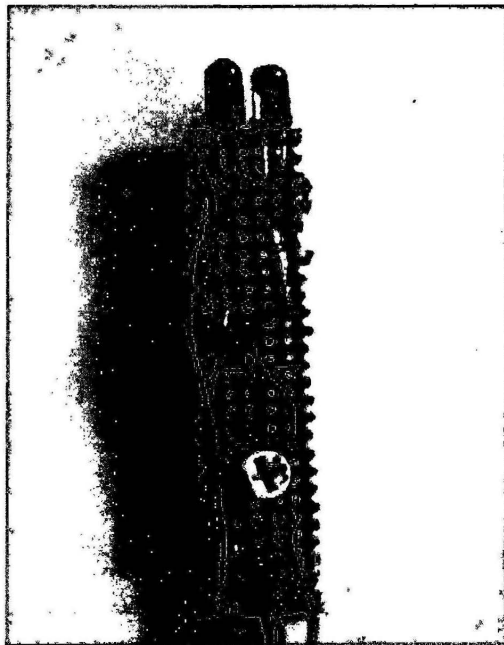


Figure 3.5: PIC18F4550 Device

The robot will use two of the 8-bit analog to digital converters to extract analog data from both the IR sensor. Outputs for high and low will be used to drive both of the DC motors.

### 3.1.4 Infrared Sensor

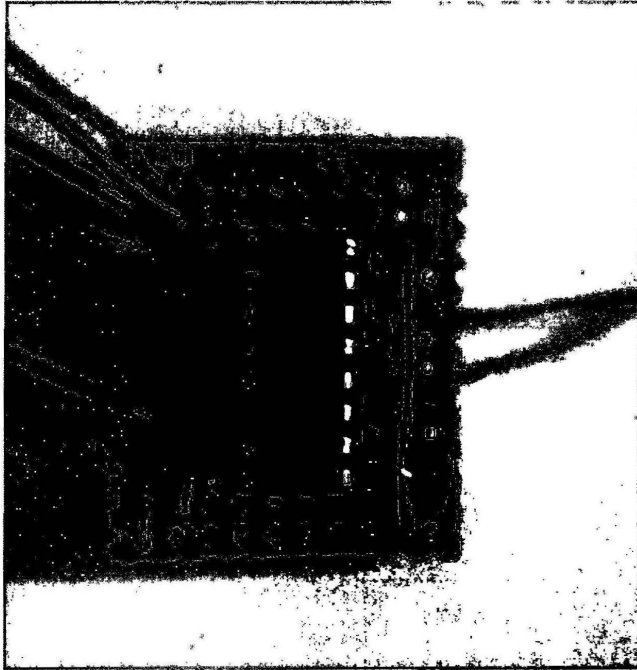
There are two pairs of infrared sensors are used in the autonomous robot. Each pair consist of one IR emitter and receiver. The sensors are placed at left and right side to detect the obstacle. Each IR sensor output will be connecting to Analog to digital pin converter from PIC18F4550. So the analog output from IR sensor will be converting into digital form and read by controller using LCD screen. In same time the controller will also determine the step to be done to move the motor direction from the command have been program into chip. The IR detecting range is around 0-12cm which enable the robot to detect the obstacle before crashes them. The range cannot be set at too far because it will cause the IR sensor to sensitive to surrounding. The potentiometer can be using to adjust the sensitivity of the IR sensor. Figure 3.6 show the IR sensor and 3.7 shows the schematic diagram of IR sensor



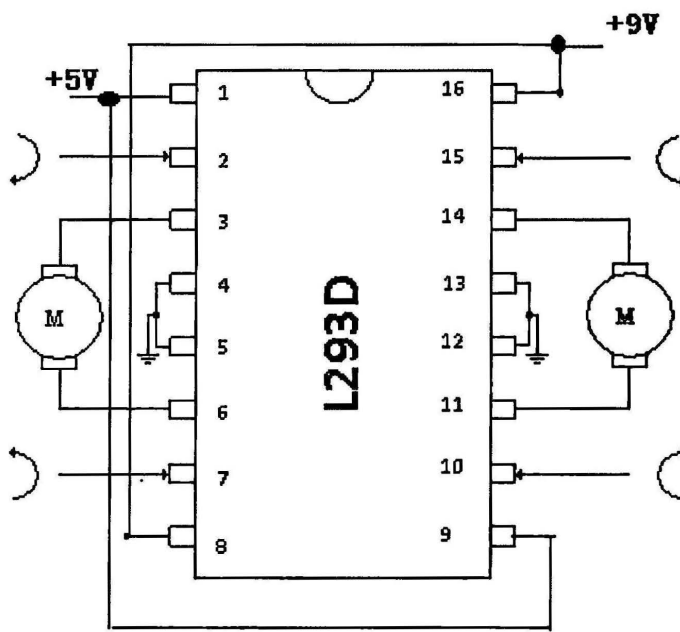
**Figure 3.6:** IR sensor







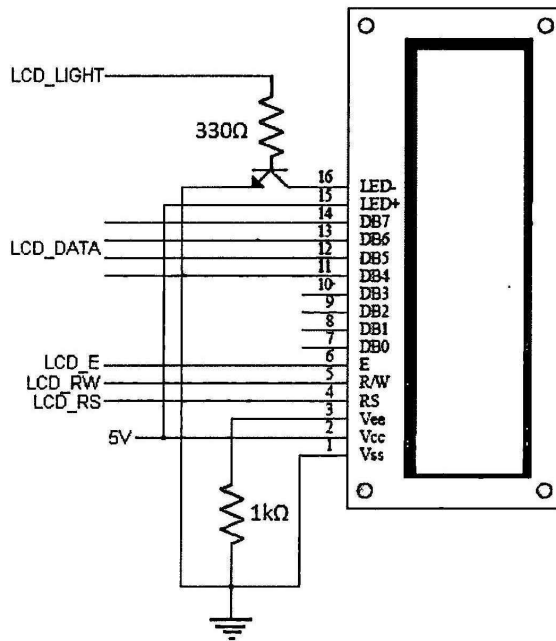
**Figure 3.8 :** L293D



**Figure 3.9:** Schematic diagram of L293D

### 3.1.6 2X16 Liquid crystal display

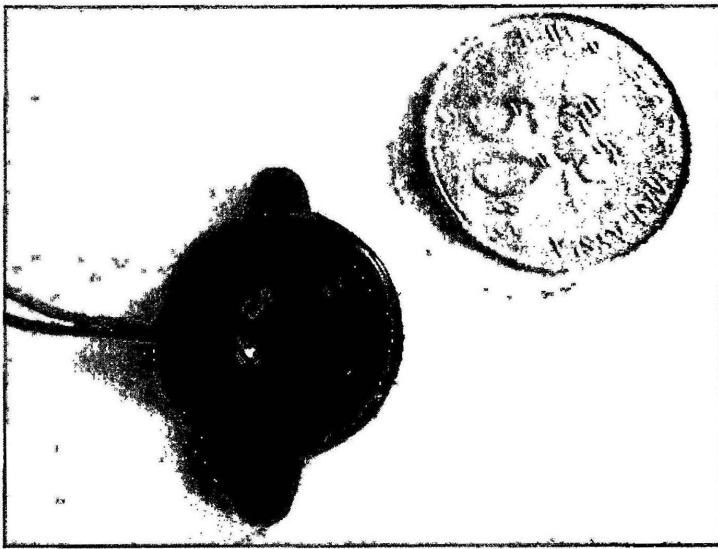
The Liquid crystal display (LCD) come in two main types that are character LCD display and pixel and also graphic LCD display. 2X16 LCD display will have 2 row of display that each row can show 16 characters. The 2x16 LCD are comes with one row sixteen pins. The first fourteen pins are used to control the display and the last two are for the backlighting. The pin 1,5and 16 of LCD is connected to ground, pin2 and 15 is connected to +5 V. The pin 4(RS) is connected to port B4 of microcontroller while pin6 (E) is connect to portB5 .For pin 7.8.9.10.11.12.13.14 (Data pins) will be connect to port D0,1,2,3,4,5,6,7.The 2X16 LCD diagram will be show in Figure 3.10.



**Figure 3.10 :** Schematic diagram of LCD

### 3.1.7 Piezo Electric Buzzer

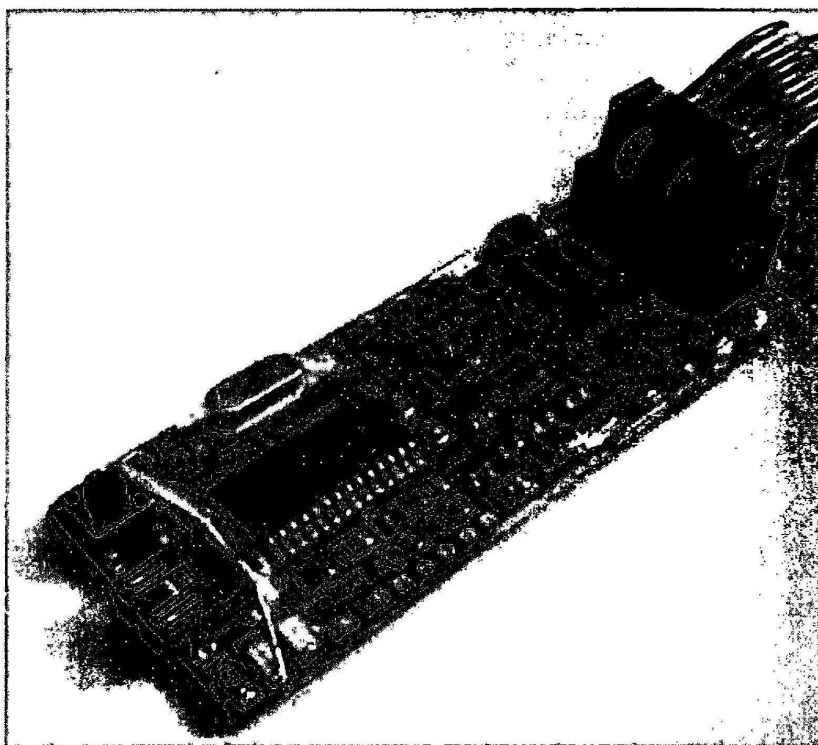
The buzzer is an audio signaling device. Piezo electric buzzers with sources is been chosen and it has the drive circuit already,only the DC voltage is needed to produce the sound. The voltage determines the loudness of this kind of products. Higher is the voltage, louder is the sound and bigger is the corresponding current. The advantages of buzzer are of consume less current, by using minimum voltage can generate loud sound output and it also can be set up easily. The Anode of buzzer is connect to the port B6 (pin39) of microcontroller the high and low value will be sent to control the buzzer on/off. Figure 3.11 show the piezo electric buzzer.



**Figure 3.11 : Piezo Electric Buzzer**

### 3.1.8 UIC00B USB ICSP PIC Programmer

UIC00B is a programmer designed to program popular Flash PIC Microcontroller which includes most of the PIC family. It can program 8bit, 16bit and 32bit PIC MCU. On board ICSP (In Circuit Serial Programming) connector offers flexible methods to load program.USB port is commonly available and widely used on Laptop and Desktop PC, thus is very convenient to use UIC00B.It compatible with Microchip's PICKit 2. It powered directly from USB port and no external power required for UIC00B to function. UIC00B also supports on-board programming which eliminates the need of plug-in and plug-out of PIC MCU ,this allows to modify the program without removing the PIC from the development board.



**Figure 3.12:** UIC00B USB ICSP PIC Programmer

## 3.2 MECHANICAL DESIGN

### 3.2.1 Dc Geared Motor

The SP30-300K dc geared motor is been chosen to drive the robot. It have specification of max 12volt and having the speed of 12 RPM, 300:1 gear ratio, current at 410mA and also 1176mN.m torque. This dc geared motor is chosen because the speed of the robot have low speed and also the higher torque value compare to other SP30 series geared motor.

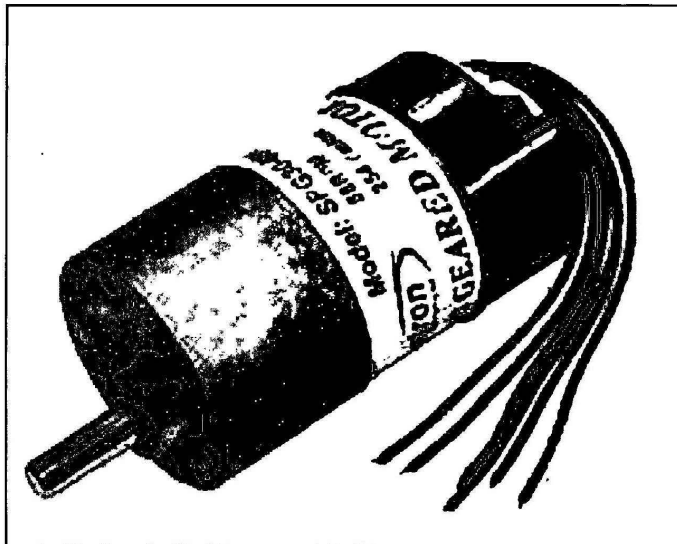


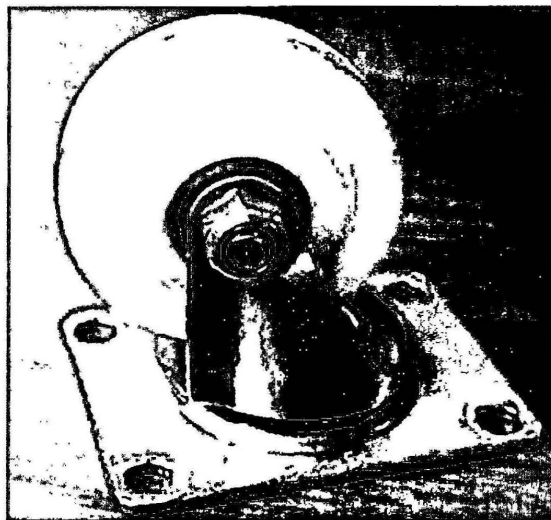
Figure 3.13: SP30-300K dc geared motor

### 3.2.2 Tires

In order to move the robot two toy truck tires are installed to the dc geared motor. Figure 3.22 shows the tires. And then the swivel caster is attaching to support chassis of the robot from scratching the floor. Figure 3.23 show the swivel caster.



**Figure 3.14 :** Toy trucks tires



**Figure 3.15:** Swivel caster

### 3.2.3 Chassis of The Robot

In order to reduce cost, an idea of using recycle material as the chassis of the robot has been suggest. Therefore from survey and founds the biscuit can is chosen as the body of the robot. The reason to choosing the biscuit can is firstly it can be easily to obtain, it is large enough to attach the dc motor and the body of the biscuit can be easily to be shaped and the controller board can be put inside of the can.



**Figure 3.16:** Chassis with dc motor, caster and tires

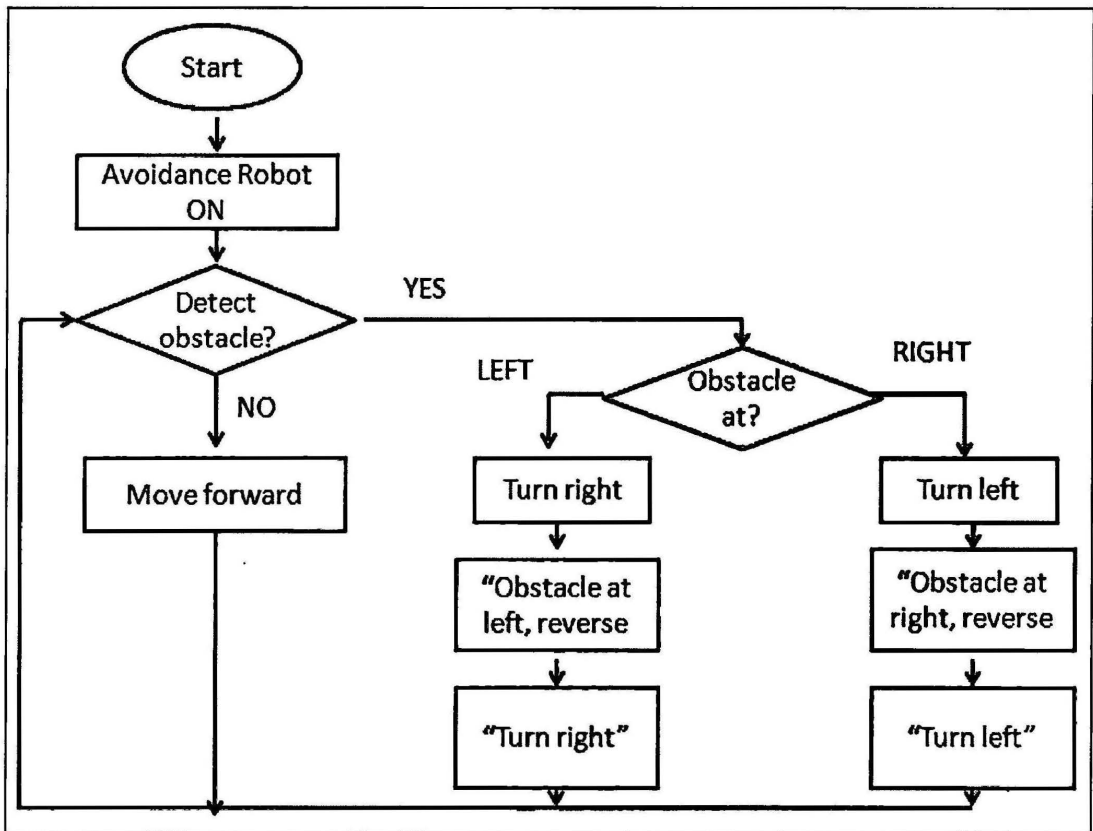
## 3. 3 SOFTWARE DESIGN

### 3.3.1 Background

Without software development, there is no use of establishing any project or robot. In software development there is a need in programming the microcontroller. In other words, programming the microcontroller is like inserting intelligence into the robot's brain. In this project the CCS compiler will be used to do the programming and it will convert to Hex file that can be read by PIC kit 2 programmers to program the programming into microchip.

### 3.3.2 Algorithm

Obstacle avoidance robot algorithm show as below will be programmed into the microcontroller unit. When the robot is ON, it will show the “Autonomous robot ON”, on the LCD screen and the buzzer will beep for 1 times to show that the robot is turning ON, the robot will moves forward if there are no obstacles. If the obstacle is detected on left site it will turn the robot to left site and show “Obstacle at left, reverse”, and “Turn right”, when detected on right side show “Obstacle at right, reverse”, and “Turn left”, Figure 3.31 show the robot algorithm.



**Figure 3.17 :** Obstacle Avoidance Robot algorithm



### 3.3.3 PCW IDE Compiler

The PCW IDE Compiler is developed by Custom Computer Services.Inc. The PCW IDE Compiler provides a complete integrated tool suite for developing and debugging embedded applications running on Microchip PIC family

The C language was use in CCS C compiler, is much easier to read, write or understand the compiler language compare to other types of compiler. The CCS compiler provides a lot of built-in functions to access and use the pic microcontroller's peripherals. This makes it very easy for the users to configure and use the peripherals without going into in depth details of the registers associated with the functionality.

To start the PCW IDE Compiler, fist create a new file at file menu, create a source file saves, In this project PIC 18F4550 will be used so the header type of PIC type will be choose at Option , project option and the target PIC18F4550 will be chosen. The header files of PIC 18F4550 can be call from library directory during writing programming.

After finish writing the program, the file need to be save in .bak first .Then only can compile the program, compilation will be terminate and error will be listed if there is error in the program. The .hex file will be generated after the compilation. The. hex file will be programmed to PIC 18F4550 after that.

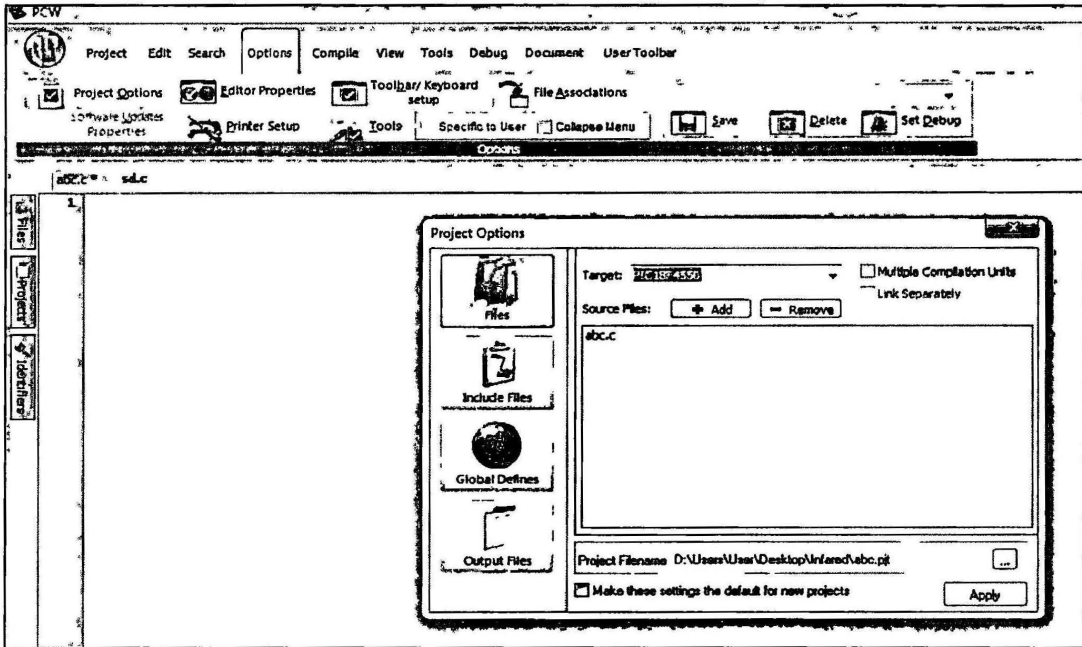


Figure 3.18: Selecting PIC type source

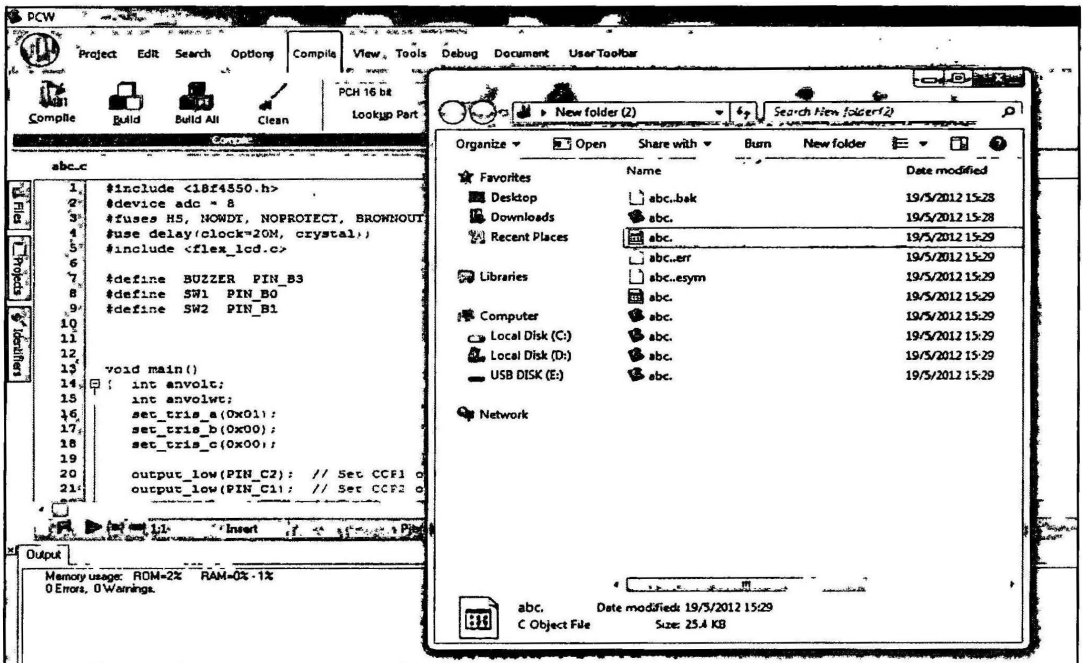


Figure 3.19: Compilation of programming and the file type generate

### 3.3.4 PICkit 2 Programmer

To program the .hex file into PIC18F4550, pickit 2 programmer is used. The UIC00B will be connect to ICSP of sk40C board and also to pc using usb connecter. Wait the device of PIC18F4550 been detected by pickit 2 programmer then go File , Import hex file, then the hex file of programming will be chosen and then click the write to program the .hex file into the microchip.

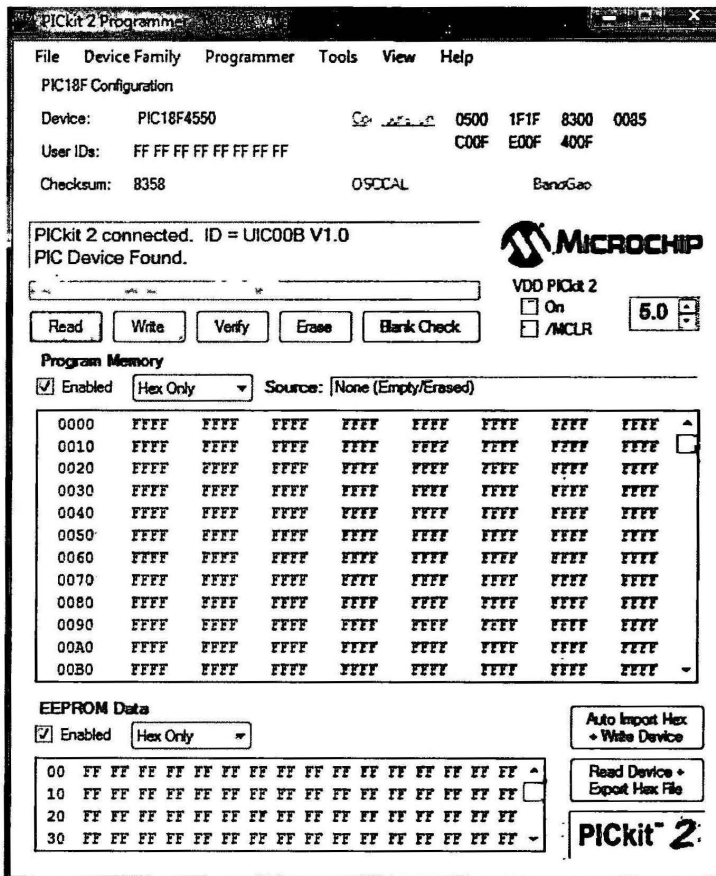


Figure 3.20: Detecting Device

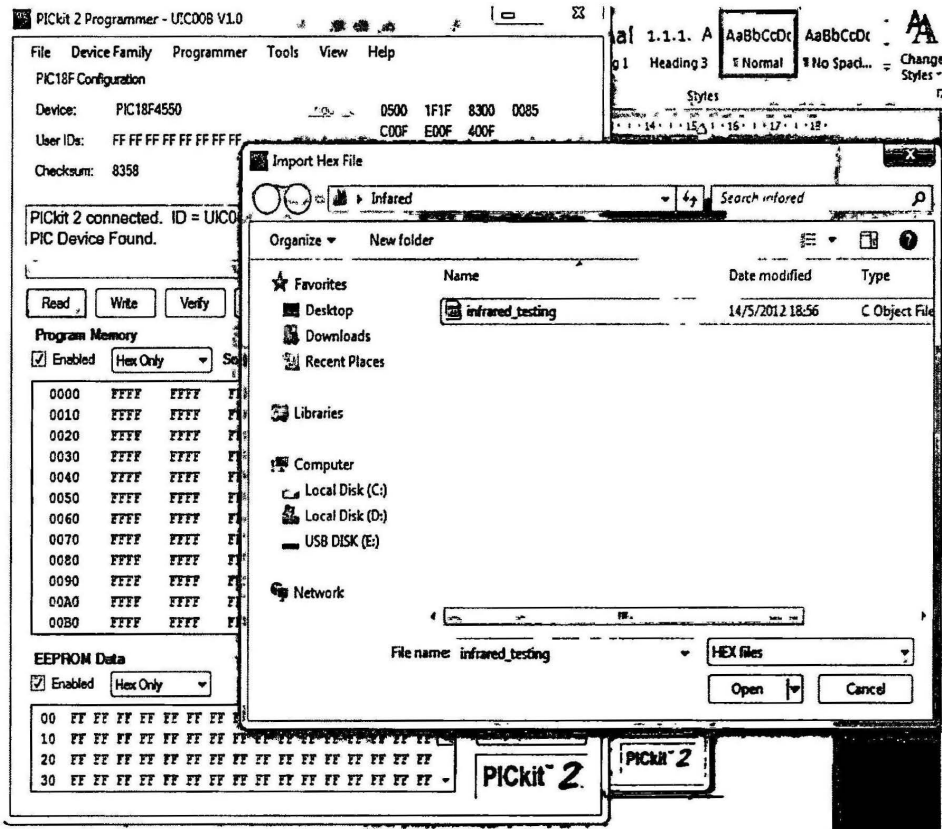


Figure 3.21: Selecting source of HEX file

**Table 4.1** : Logic and voltage values of L293D when robot moving forward

SENT	INPUT	VOLTAGE	OUTPUT	VOLTAGE
HIGH	1	4.9	1	4.9
LOW	2	0.04	2	0.04
HIGH	3	4.9	3	4.9
LOW	4	0.04	4	0.04

The input 1 and 3 been sent HIGH value gave the voltage 4.9V show that logics “1” are sent by portB0 and portB2 to input L293D ,while input 2 and 4 are getting 0.04V this show that logics “0” are sent by portB1 and portB3. The result match the program. For output 1 and 2 been show 4.9V and 0.04 V measuring by using multi meter. The voltage difference between two output let the left motor rotating forward and this also same with output 3 and 4 that getting 4.9V and 0.04 V .Both motor rotating forward ,which mean the robot also move forward..

#### 4.1.2 Robot Reverse

In order to let the robot to reverse, mean both of the motors must rotate backward. To let the left motor rotates backward, LOW will be sent by portB0 to input1, HIGH by portB1 to input2, LOW by portB2 to input3, HIGH by port B3 to input 4. The programming are show as bellow.

```
output_low (B0);
output_high (B1);
output_low (B2);
output_high (B3);
```

**Table 4.2 :** Logic and voltage values of L293D when robot moving reverse

SENT	INPUT	VOLTAGE	OUTPUT	VOLTAGE
LOW	1	0.04	1	0.04
HIGH	2	4.9	2	4.9
LOW	3	0.04	3	0.04
HIGH	4	4.9	4	4.9

The input 2 and 4 been sent HIGH value gave the voltage 4.9V show that logics “1” are sent by portB0 and portB2 to input L293D ,while input 1 and 3 are getting 0.04V this show that logics “0” are sent by portB1 and portB3. The result match the program. For output 1 and 2 been show 0.04V and 4.9 V measuring by using multi meter. The voltage difference between two output let the left motor rotating backward and this also same with output 3 and 4 that getting 0.04V and 4.9 V. Both motor rotating backward, which mean the robot also reverse.

#### 4.1.3 Robot Turn Left

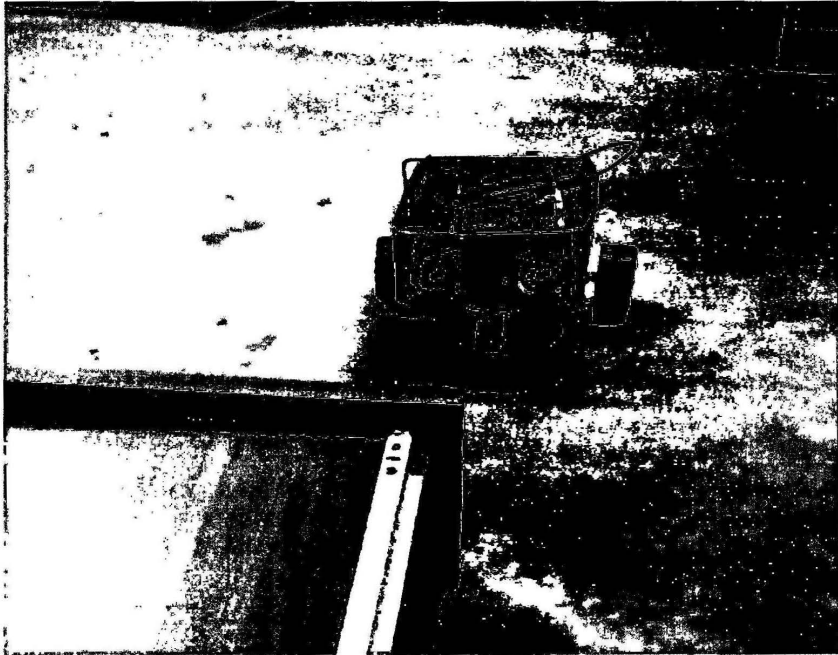
In order to let the robot to left, the left motors must rotate backward and right motor must move forward. To let the left motor rotates backward, LOW will be sent by portB0 to input1, HIGH by portB1 to input2, for right motor to move forward HIGH sent by portB2 to input3 , LOW port B3 to input 3. The programming is show as bellow.

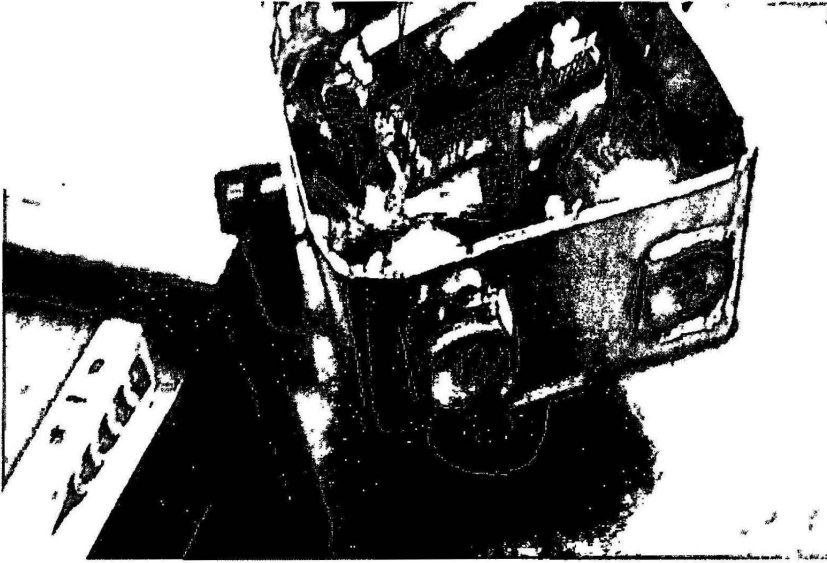
```
output_low(B0);
output_high(B1);
output_high(B2);
output_low(B3);
```

**Table 4.3 :** Logic and voltage values of L293D when robot moving left

SENT	INPUT	VOLTAGE	OUTPUT	VOLTAGE
LOW	1	0.04	1	0.04
HIGH	2	4.9	2	4.9
HIGH	3	4.9	3	4.9
LOW	4	0.04	4	0.04

The input 2 and 3 been sent HIGH value gave the voltage 4.9V show that logics "1" are sent by portB1 and portB2 to input L293D ,while input 1 and 4 are getting 0.04V this show that logics "0" are sent by portB0 and portB3.The result match the program. For output 1 and 2 been show 0.04V and 4.9 V measuring by using multi meter. The voltage difference between two output let the left motor rotating backward while for output 3 and 4 getting 4.9V and 0.04 V that give motor move forward .Left motor moving backward and right motor moving forward let the robot turn left.

**Figure 4.1:** Obstacle at right



**Figure 4.2:** Turn left

#### **4.1.4 ROBOT TURN RIGHT**

In order to let the robot to right, the left motors must rotate forward and right motor must move backward. To let the left motor rotates forward, high will be sent by portB0 to input1, low by portB1 to input2, for right motor to move backward LOW sent by portB2 to input3 , HIGH port B3 to input 3. The programming is show as bellow.

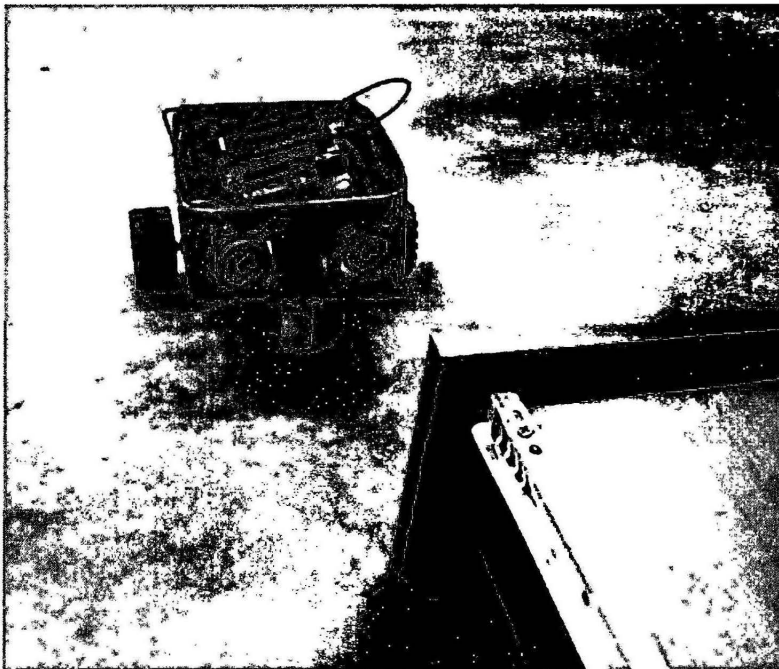
```
output_high (B0);  
output_low(B1);  
output_low(B2);  
output_high(B3);
```

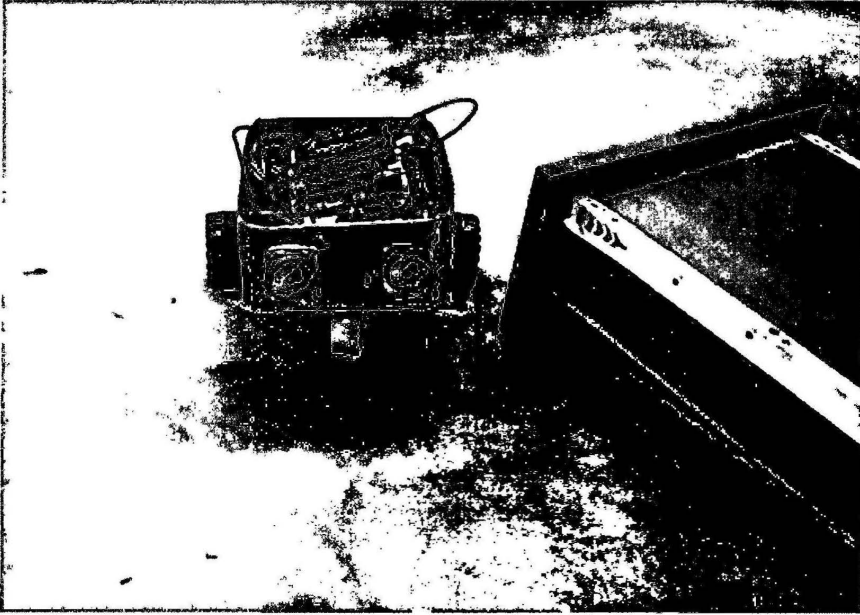


**Table 4.4 :** Logic and voltage values of L293D when robot moving right

SENT	INPUT	VOLTAGE	OUTPUT	VOLTAGE
HIGH	1	4.9	1	4.9
LOW	2	0.04	2	0.04
LOW	3	0.04	3	0.04
HIGH	4	4.9	4	4.9

The input 1 and 4 been sent HIGH value gave the voltage 4.9V show that logics “1” are sent by portB0 and portB3 to input L293D ,while input 2 and 3 are getting 0.04V this show that logics “0” are sent by portB1 and portB2.The result match the program. For output 1 and 2 been show 4.9 and 0.04 V measuring by using multimeter. The voltage difference between two output let the left motor rotating forward while for output 3 and 4 getting 0.04V and 4.9V that give motor move backward .Left motor moving backward and right motor moving forward let the robot turn left.

**Figure 4.3:** Obstacle at left



**Figure 4.4:** Turn right

## 4.2 IR EMITTER AND RECEIVER TESTING

After IR sensor been build. It will be tested by measuring the voltages value of the receivers' output. When the obstacle is detected the voltage output of IR sensor will increase, the more close the obstacle to IR sensor the bigger voltage value can get. The potentiometer is use to adjust the sensitivity of IR sensor.

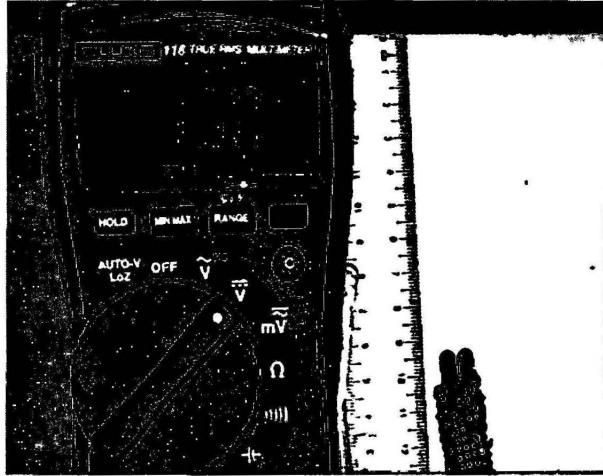


Figure 4.5 : Obstacle at distance 9cm

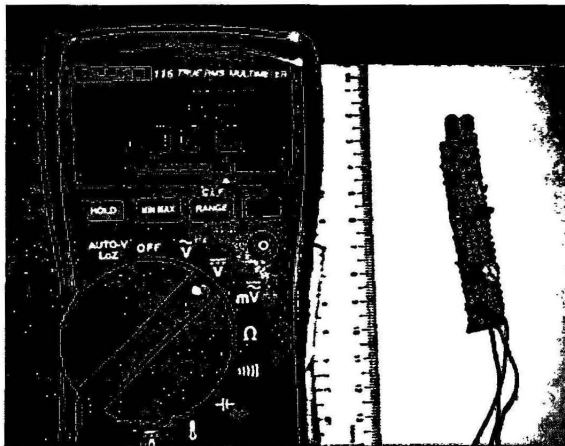
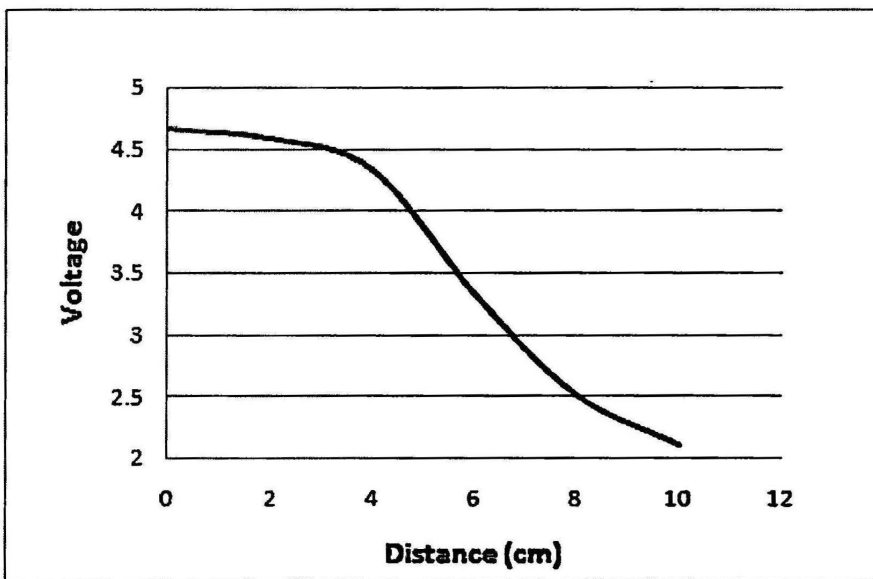


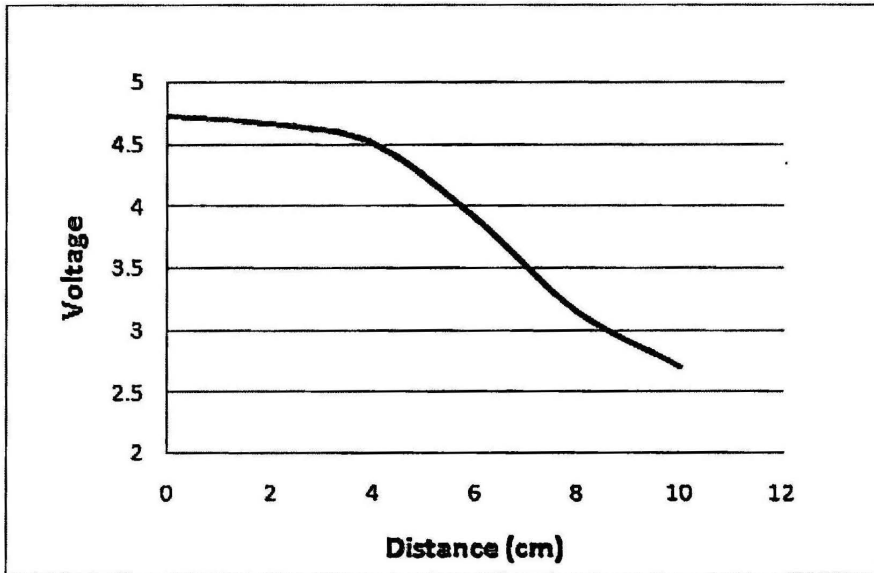
Figure 4.6: Obstacle at distance 2cm

**Table 4.5 :** Distance and voltage value for left IR sensor

Distance(cm)	Voltage
0	4.66
2	4.58
4	4.33
6	3.34
8	2.52
10	2.1

**Figure 4.7:** Graph of Voltage versus Distance for left IR sensor**Table 4.6:** Distance and voltage value for right IR sensor

Distance(cm)	Voltage
0	4.73
2	4.66
4	4.5
6	3.9
8	3.15
10	2.7

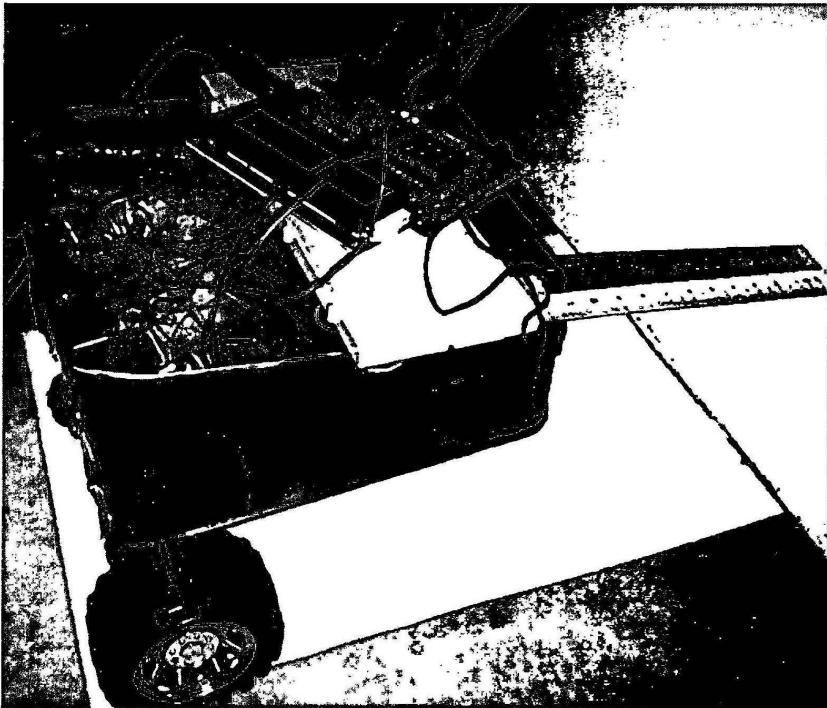


**Figure 4.8:** Graph of Voltage versus Distance for right IR sensor

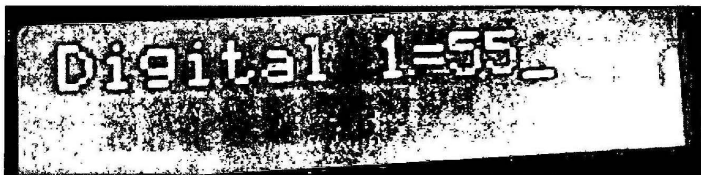
From the result measure can found out that right IR sensor have the higher output voltage value compare to left IR sensor, but the graph of both IR sensor is almost alike. The value of difference in voltage among both sensors is cause by the small difference value of resistance among the IR sensor. The IR sensor may be too sensitive to surrounding if the distance to measure is too far, so the potential meter that build in the IR sensor can be use to adjust the voltage value/distance. And also from the graph can found that voltage output of IR sensor can found that from distance 4cm- distance 8 cm voltage can be found direct proportional to distance. So in this prove that the distance around 4-8cm can be choosing for idea distance for IR sensor to detect the obstacle.

### 4.3 ADC

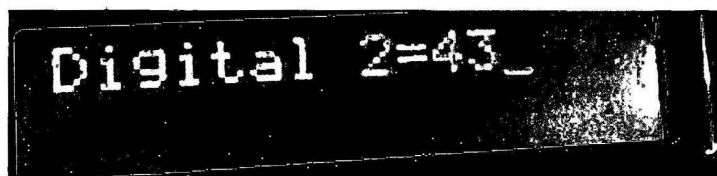
The Analog to digital converter is used to convert an analog signal to digital number from IR sensor in this project. From the digital value that has been recorded an equation will be built to use in the programming to help the microcontroller determine the distance of the obstacle. 8 bit ADC is being used. Preset/potential meter are used to adjust sensitivity and get the suitable graph to build equation for programming.



**Figure 4.9:** Get Digital value for IR left and right using ADC



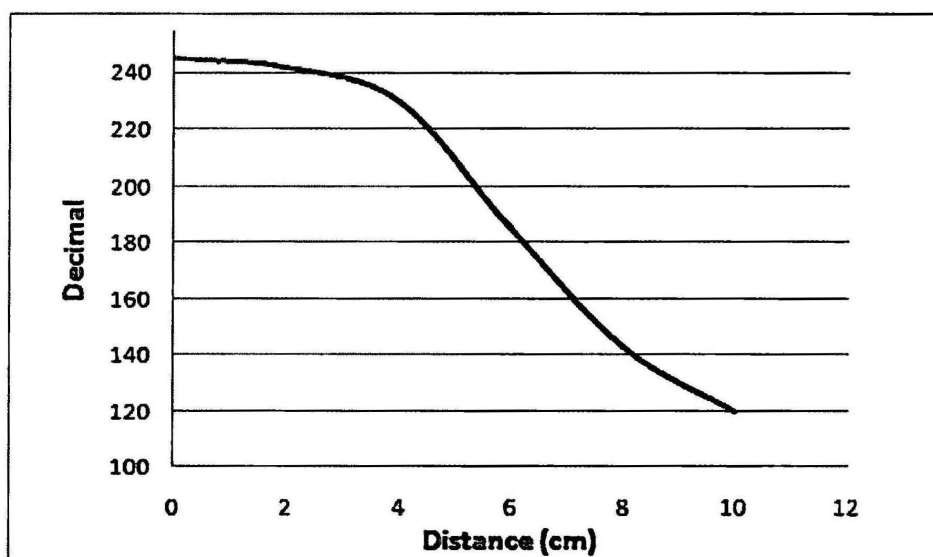
**Figure 4.10:** Sample of Decimal value for left sensor



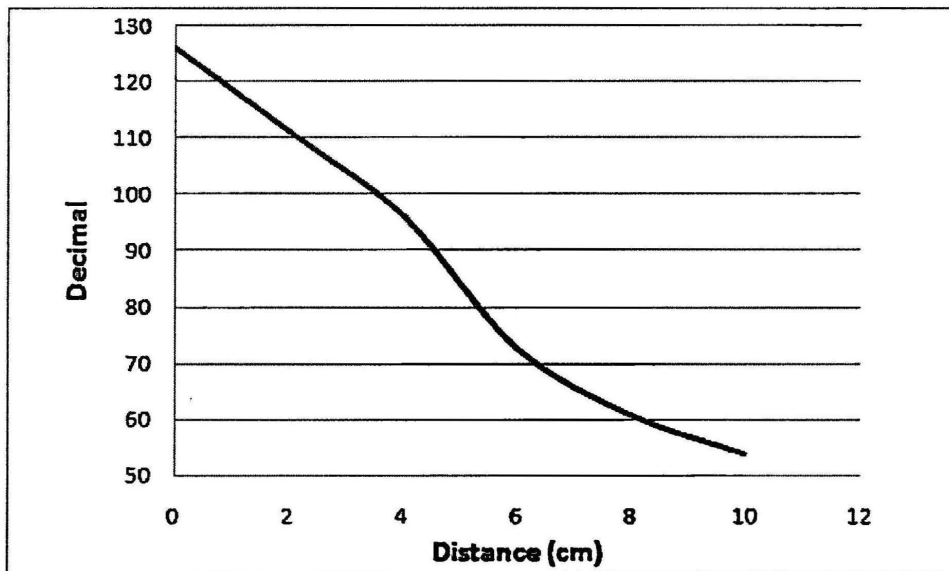
**Figure 4.11:** Decimal values for right sensor

**Table 4.7 :** Distance and decimal value for left IR sensor before ( $R_2=49\text{kohm}$ ) and after adjust ( $R_2=20\text{kohm}$ ) potentiometer ( $50\text{kohm}$ )

Distance(cm)	Equivalent decimal value	Distance(cm)	Equivalent decimal value
0	245	0	126
2	242	2	111
4	230	4	96
6	185	6	73
8	143	8	61
10	120	10	54
Before adjust		After Adjust	



**Figure 4.12:** Decimal versus Distance left IR sensor before adjust potentiometer



**Figure 4.13:** Decimal versus Distance left IR sensor after adjust potentiometer

From the result that we get for decimal value for left IR sensor, the graph that before adjust the preset, the decimal point slightly decreasing while the distance is slightly increasing. The graph that gets after the adjustment of the preset, is almost decrease directly proportional. The decimal value is decreasing directly proportionally to the distance after the preset adjusted. The better graph get after the sensitivity of the IR sensor was reduce a bit. The linear equation will be form below Eq.(4.1)(4.2)(4.3)

$$Y = mX + C \quad (4.1)$$

The gradient  $m$  is the calculated using measurement table 4.7

$$\frac{54 - 126}{10 - 0} = -7.2$$

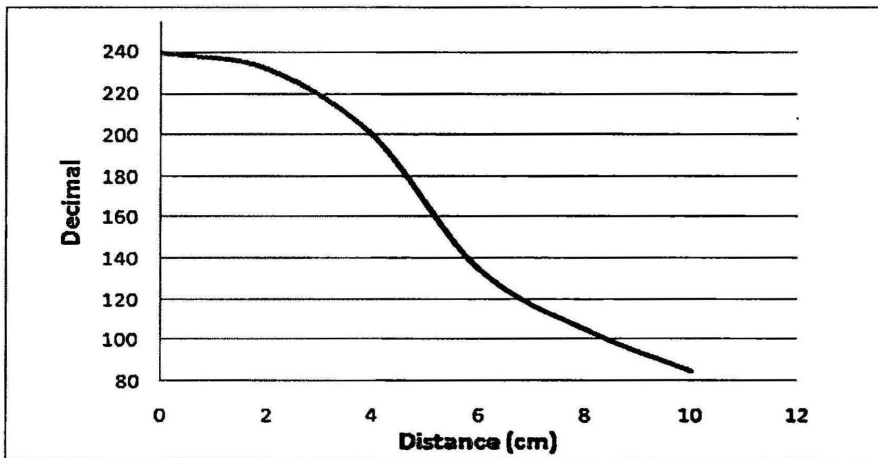
$$Y = -7.2X + 126 \quad (4.2)$$

Where  $Y$  is the decimal value read by ADC,  $X$  is the distance of the obstacle from robot.

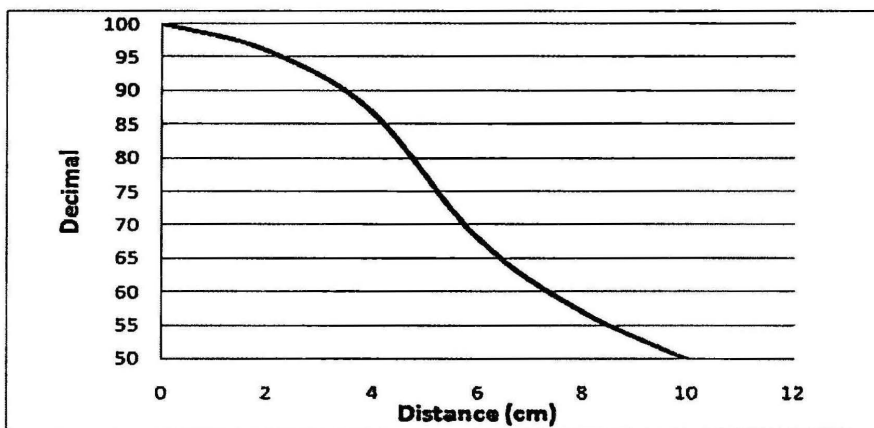


**Table 4.8:** Distance and decimal value for right IR sensor before ( $R_2=49\text{kohm}$ ) and after adjust ( $R_2=20\text{kohm}$ ) potentiometer ( $50\text{kohm}$ )

Distance(cm)	Equivalent decimal value	Distance(cm)	Equivalent decimal value
0	240	0	100
2	232	2	96
4	199	4	87
6	135	6	68
8	105	8	57
10	85	10	50
Before adjust		After Adjust	



**Figure 4.14:** Decimal versus Distance right IR sensor before adjust preset



**Figure 4.15:** Decimal versus Distance right IR sensor after adjust preset

From the result that we get for decimal value for right IR sensor, the graph that before adjust the preset, the decimal point slightly decreasing while the distance is slightly increasing. The graph that gets after the adjustment of the preset, is almost decrease directly proportional. The decimal value is decreasing directly proportionally to the distance after the preset adjusted. The better graph get after the sensitivity of the IR sensor was reduce a bit. The linear equation will be form below.

$$Y = mX + C \quad (4.3)$$

The gradient  $m$  is the calculated using measurement table 4.8

$$\frac{50-100}{10-0} = -5.0$$

$$Y = -5X + 100 \quad (4.4)$$

Where  $Y$  is the decimal value read by ADC,  $X$  is the distance of the obstacle from robot

#### 4.4 LIMITATION OF THE ROBOT

By observing the result and the testing of the project, there is some limitation need to consider in this project. Firstly is the battery duration for the robot, the 9V battery is used as the power supply of robot, the battery will be easily exhausted during movement of robot, and also LCD backlight will also directly reduce the duration of batteries lifetimes. The front wheel of robot also will sliding during the turning of direction of robot by using the controlling of back wheel., the direction control of robot is suggested to put in front to avoid the sliding of wheel that affect the direction of robot moving. Another limitation is IR sensor, it will be easily disturb by sunlight in outdoor, and the sensitivity need to adjust by potentiometer if got exist of direct or indirect sunlight. Therefore ultra sonic sensor is recommended to use to overcome the disturbance by sunlight but the cost will be increase because **ultra sonic sensor** is expensive.

## CHAPTER5

### CONCLUSION

#### 5.1 CONCLUSION

As a conclusion, this obstacle avoidance robot achieves the objective after the integration is done while it also fulfills the scopes of this project. The obstacle avoidance robot can be function well as been planned. In future a better functionary obstacle avoidance robot can be built; the improvement can be done by adding more expensive and high technology gadget such as wireless control, ultra sonic sensor and so on.



**Figure 5.1:** Complete Obstacle Avoidance Robot

## 5.2 FUTURE DEVELOPMENT

Future work can also be implemented to enhance this autonomous function:

- I. Replacing the IR sensor with ultra sonic sensor, to overcome the sunlight from disturbing the detection of obstacle.
- II. Temperature sensor or humidity sensor can be added to sense the surrounding environment.
- III. More sensors can be added to improve the robot obstacle –avoidance ability.
- IV. The solar panel can be added as the replacement/second power supply for the robot.
- V. Sound chip coder can be added to play the pre record message.

## REFERENCES

- [1]. Asimov I (1950) I, Robot. Fawcett, Greenwich, CT.
- [2] Asimov I (1942) Runaround. In: Astounding science Fiction. Street & Smith Publications Inc., New York.
- [3] Zhejun Fan\* Johann Borstlein\* David Wehe\*\* Yoram Koren\* , “Experimental Evaluation of an Encoder Trailer for Dead-reckoning in Tracked Mobile Robots .”, 0-7803-2722-5/95 1995 IEEE m/s571-576.
- [4] Rui Araujo and Anibal T. de Almeida,, “Learning Sensor-Based Navigation of a Real Mobile Robot in Unknown Worlds.” IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 29, NO. 2, APRIL 1999,pp164-178.
- [5] Reza Abbaspour, “Design and Implementation of Multi-sensor Based Autonomous Minesweeping Robot.” 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 978-1-4244-7286-4/10 2010 IEEE,pp443-449.
- [6] U. Remhold, T. Lueth, and T. Ogasawara . “From Autonomous Assembly Robots to Service Robots for Factories”. Institute for Real-Time Computer Systems and Robotics IPR ,University of Karlsruhe, D-76128, Karlsruhe.
- [7] Dror G. Feitelson. “Asimov’s Laws of Robotics Applied to Software”.IEEE SOFTWARE, pp112-113, July / August 2007.
- [8] Roger Clarke, “Asimov’s Laws of Robotics: Implications for Information Technology” Australian National University, January 1994 .

- [9] Hongbo Wang, "Development of Motor Controller Based on PIC," 2007 International Conference on Convergence Information Technology 0-7695-3038-9/07 \$25.00 © 2007 IEEE .
  
- [10] David Meiklejohn, "Introduction to PIC Programming," Gooligum Electronics 2008, pp1-8.
  
- [11] NORMAN S.NISE ,Control System Engineering sixth edition ,Wiley 2011, pp7-9.
  
- [12] G. Welch and G. Bishop Anca Apatean, Alexandrina Rogozan, and Abdelaziz Bensrhair, "Obstacle recognition using multiple kernel in visible and infrared images." 978-1-4244-3504-3/09/2009 IEEE.

**APPENDIX A**  
**Obstacle Avoidance Robot Programming**



```

/*****
Project : Avoidance Robot
Author : LEE LE XIN
Date : 27/4/2012
*****/
#include <18f4550.h>
#define adc = 8
#define HS, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
#define use delay(clock=20M, crystal)
#include <flex_lcd.c> // Call the LCD program

#define BUZZER PIN_B6
#define MTL1 PIN_B0
#define MTL2 PIN_B1
#define MTR1 PIN_B2
#define MTR2 PIN_B3

void main()
{ int YL;
  int YR;
  float XL;
  float XR;

  YL=(-7.2)*XL+126 ;
  YR=(-5)*XR+100 ;

  set_tris_a(0b11111111);
  set_tris_b(0x00000000);
  set_tris_c(0x00000000);

  setup_adc_ports(NO_ANALOGS);
  setup_adc(ADC_CLOCK_INTERNAL);
  lcd_init();
  printf(lcd_putc,"Autonomous Robot ");
  delay_ms(2000);
  lcd_init();
  printf(lcd_putc,"ON ");
  delay_ms(2000);

  output_high (BUZZER);
  delay_ms(2500);
  output_low (BUZZER);

  output_high ( MTL1); // forward
  output_low ( MTL2);
  output_high (MTR1);
  output_low(MTR2);

```

```

while(true)
{

    output_high ( MTL1);    // forward
    output_low ( MTL2);
    output_high (MTR1);
    output_low(MTR2);

    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(5);
    delay_us(500);

    YL = read_adc(5);
    lcd_init();
    printf(lcd_putc,"Distance 1=%u",YL);
    delay_ms(1500);

    read_adc(ADC_START_ONLY);

    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(4);
    delay_us(500);

    YR = read_adc(4);
    lcd_init();
    printf(lcd_putc,"Distance 2=%u",YR);
    delay_ms(2000);
    read_adc(ADC_START_ONLY);

if(YL>=60)
{
    output_low ( MTL1);    // backward
    output_high ( MTL2);
    output_low (MTR1);
    output_high (MTR2);
    lcd_init();
    printf(lcd_putc, "Obstacle,left" );
    delay_ms(2200);

    lcd_init();
    printf(lcd_putc, "Reverse" );
    delay_ms(2200);
}
}

```

```

    output_high ( MTL1);    //turn right
    output_low( MTL2);
    output_low (MTR1);
    output_high(MTR2);

    lcd_init();
    printf(lcd_putc, "Turning right");
    delay_ms(2200);

}

else if(YR>=60)

{
    output_low ( MTL1);    // backward
    output_high ( MTL2);
    output_low (MTR1);
    output_high (MTR2);
    lcd_init();
    printf(lcd_putc, "Obstacle,right" );
    delay_ms(2200);

    lcd_init();
    printf(lcd_putc, "Reverse ");
    delay_ms(2200);
    output_low ( MTL1);    //turn left
    output_high( MTL2);
    output_high (MTR1);
    output_low (MTR2);

    lcd_init();
    printf(lcd_putc, "turning left");
    delay_ms(2200);

}

}
}

```

```

/*****
**
Project : Avoidance Robot
Author : LEE LE XIN
Date : 20/5/2012
*****/
#include <18f4550.h>
#define adc = 8
#fuses HS, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
#use delay(clock=20M, crystal)
#include <flex_lcd.c> // Call the LCD program

#define BUZZER PIN_B6
#define MTL1 PIN_B0
#define MTL2 PIN_B1
#define MTR1 PIN_B2
#define MTR2 PIN_B3

void main()
{ int YL;
  int YR;
  float XL;
  float XR;
  set_tris_a(0b11111111);
  set_tris_b(0b00000000);
  set_tris_c(0b00000000);

  setup_adc_ports(NO_ANALOGS);
  setup_adc(ADC_CLOCK_INTERNAL);
  lcd_init();
  printf(lcd_putc,"Avoidance Robot ");
  delay_ms(1000);
  lcd_init();
  printf(lcd_putc,"ON ");
}

```

```
delay_ms(1000);

output_high (BUZZER);
delay_ms(1500);
output_low (BUZZER);

    output_high ( MTL1); // forward
    output_low ( MTL2);
    output_high (MTR1);
    output_low(MTR2);
while(true)
{

    output_high ( MTL1); // forward
    output_low ( MTL2);
    output_high (MTR1);
    output_low(MTR2);

setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_CLOCK_INTERNAL);
set_adc_channel(5);
delay_us(400);

    YL = read_adc(5);
    lcd_init();
    printf(lcd_putc,"Distance 1=%u",YL);
    delay_ms(800);

    read_adc(ADC_START_ONLY);

    setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_CLOCK_INTERNAL);
```

```
    set_adc_channel(4);
delay_us(400);

    YR = read_adc(4);
    lcd_init();
    printf(lcd_putc,"Distance 2=%u",YR);
    delay_ms(800);
    read_adc(ADC_START_ONLY);

if(YL>=80)

{
    lcd_init();
    printf(lcd_putc, "Obstacle,left" );
    delay_ms(500);
    lcd_init();
    printf(lcd_putc, "Reverse" );
    delay_ms(500);

    output_low ( MTL1);    // backward
    output_high ( MTL2);
    output_low (MTR1);
    output_high (MTR2);
    delay_ms(5000);

    lcd_init();
    printf(lcd_putc, "Turning right");
    delay_ms(500);

    output_high ( MTL1);    //turn right
    output_low( MTL2);
    output_low (MTR1);
```

```
    output_high(MTR2);
    delay_ms(5000);

}

else if(YR>=80)

{
lcd_init();
    printf(lcd_putc, "Obstacle,right" );
    delay_ms(500);
    lcd_init();
    printf(lcd_putc, "Reverse ");
    delay_ms(500);
output_low ( MTL1);    // backward
    output_high ( MTL2);
    output_low (MTR1);
    output_high (MTR2);
    delay_ms(5000);

    lcd_init();
    printf(lcd_putc, "turning left");
    delay_ms(500);
output_low ( MTL1);    //turn left
    output_high( MTL2);
    output_high (MTR1);
    output_low (MTR2);
    delay_ms(5000);

}
}
}
```

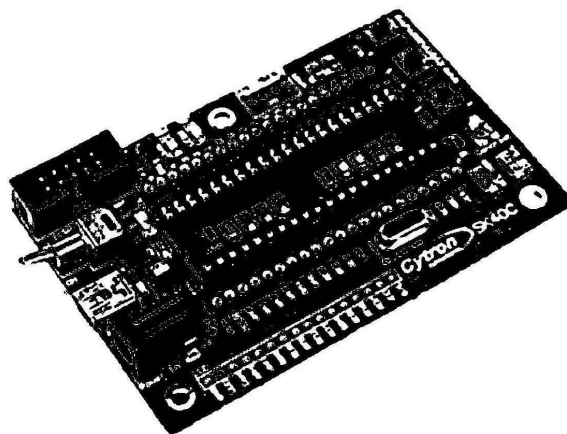
**APPENDIX B**  
**User Manual**





**SK40C**

**ENHANCED 40 PINS PIC START-UP KIT**



**User's Manual**

**V1.2**

**Dec 2010**

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

## 1. INTRODUCTION AND OVERVIEW

SK40C is another enhanced version of 40 pins PIC microcontroller start up kit. It is designed to offer an easy to start board for PIC MCU user. However, all interface and program should be developed by user. This board comes with basic element for user to begin project development. It offer plug and use features. This kit is designed to offer:

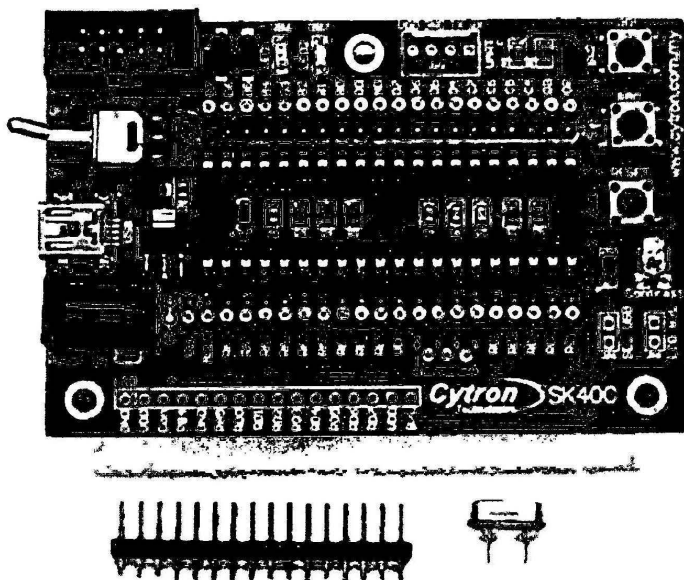
- Industrial grade PCB
- Every board is being fully tested before shipped to customer
- Compact, powerful, flexible and robust start-up platform
- Suitable for hobbyists and experts
- Save development and soldering time
- No extra components required for the PIC to function
- All 33 I/O pins are nicely labeled to avoid miss-connection by users
- Connector for UIC00A (low cost USB ICSP PIC Programmer) - simple and fast method to load program
- Fully compatible with SK40B
- No more frustrated work plugging PIC out and back for re-programming
- Perfectly fit for 40 pins 16F and PIC18F PIC
- With UIC00A, program can be loaded in less than 5 seconds
- More convenient to use and it is smaller than SK40B.
- Maximum current is 0.5A.
- Dimension: 85mm x 55mm

SK40C come with additional features:

- 2 x Programmable switch
- 2 x LED indicator
- Turn pin for crystal. User may use others crystal provided.
- LCD display (optional)
- UART communication
- USB on board.
- And all the necessities to eliminate users difficulty in using PIC

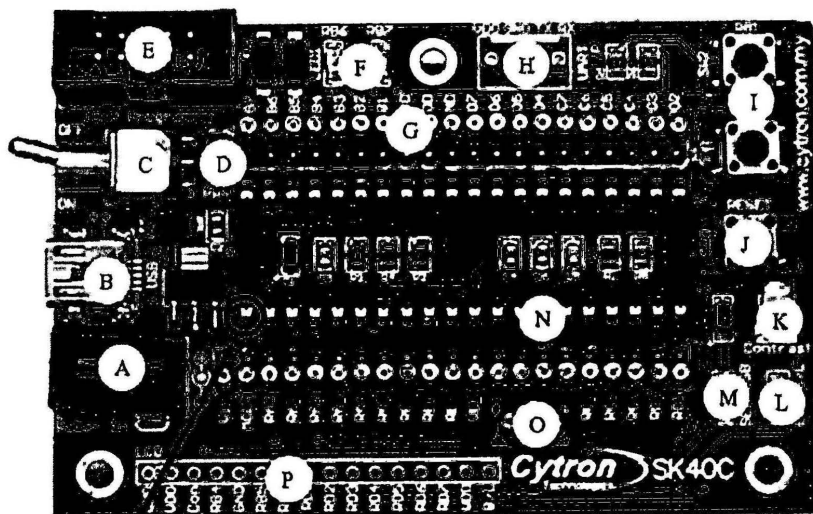
## 2. PACKING LIST

Please check the parts and components according to the packing list. If there are any parts missing, please contact us at [sales@cytron.com.my](mailto:sales@cytron.com.my) immediately.



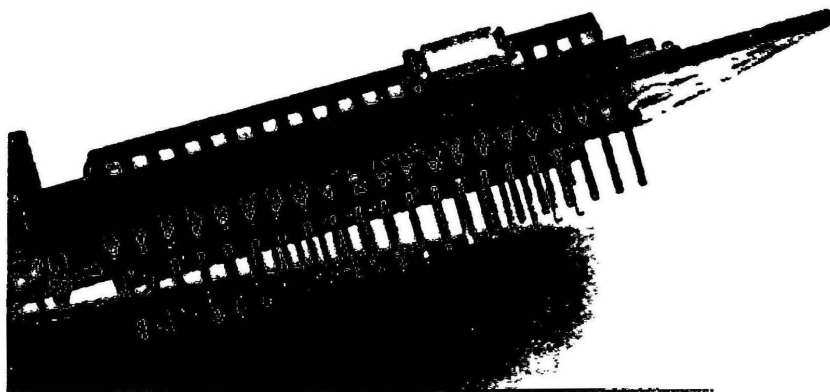
1. 1 x SK40C board with all components shown soldered
2. 1 x 16 Header pin for LCD Display.
3. 1 x 20M Hz Crystal
4. PIC MCU – Not included, please purchase separately from Cytron website
5. USB Cable – Not included, please purchase separately from Cytron website
6. UIC00A – Not included, please purchase separately from Cytron website
7. User Manual – Not included, please download from Cytron website

### 3. BOARD LAYOUT

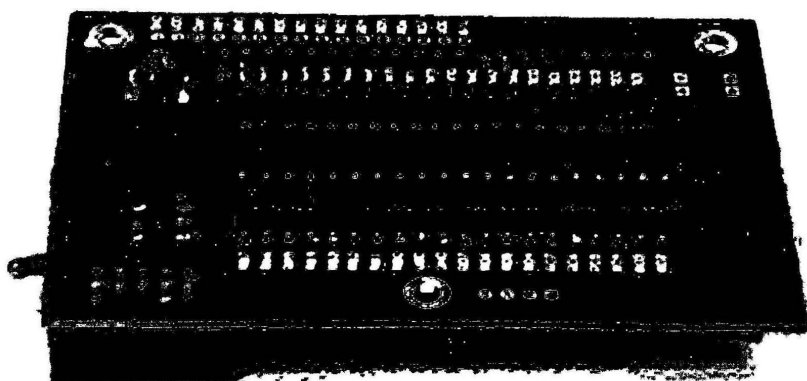


Pin 1  
of PIC

a) Top View



b) Side view



c) Bottom view

Label	Function	Label	Function
A	DC power adaptor socket	I	Programmable Push Button
B	USB Connector	J	Reset button
C	Toggle Switch for power supply	K	LCD contrast
D	Power indicator LED	L	JP8 for LCD Backlight
E	Connector for UIC00A Programmer	M	JP9 for USB
F	LED Indicator	N	40 pin IC socket for PIC MCU
G	Header pin and turn pin	O	Turn pin for crystal
H	UART Connector	P	LCD Display

**A** – DC power adaptor socket for user to plug in DC adaptor. The input voltage should be ranged from 7 to 15V.

**B** – USB connector for communication between devices and a host controller (usually personal computers). This function is only valid for certain model of PIC microcontroller. Please refer to USB interface section. The power LED will light ON when the USB cable is connected.



- 
- C** – Toggle switch to On/Off the power supply from DC adaptor.
- D** – Power indicator LED for on board. It will light ON as long as the input power is correctly connected.
- E** – 2x5 box header for UIC00A, USB ICSP PIC Programmer.
- F** – 2 LEDs (connected to RB6 and RB7) as active High output for PIC MCU. These LEDs are controllable from PIC MCU.
- G** – Consist of several line of header pin and turn pin. Header pin provide connector for user to solder SK40C to prototype board and use the I/O of PIC MCU. It is fully compatible between SK40B. Turn pin offer simple way to check voltage with multi-meter probe. 40 pins of PIC MCU except OSC (connected to crystal) are extended out to these pin. There is an extra pin on top of MCLR which is labeled as  $V_{in}$ , is connected to the input power.
- H** – Reserved for UART communication. Tx and Rx pin of SK40C are connect to RC6 and RC7 respectively. Ensure PIC use have the correct UART pin (RC6 and RC7).
- I** – 2 x Push button connected to RB0 and RB1 of PIC MCU. This is extra input button for user. It can be programmed as input switch.
- J** – Push button with the function of Reset for PIC MCU.
- K** – 5K of trimmer to set LCD contrast.
- L** – JP8 is provided for LCD Backlight. LCD Display will have backlight if this pin is shorted.
- M** – JP9 is provided for USB. Connect this pin if users use USB port
- N** – 40 pin IC socket for user to plug in any 40 pin PIC MCU (8 bit). It can either be 16F or 18F PIC. Of course the IC package should be PDIP. Please ensure the first pin is at the top side. Inside IC socket, there some electronics components, it include a 20MHz Crystal.

**O** – Turn pin is provided for crystal. 20M Hz is default crystal provided in SK40C. However, the 20M Hz crystal can be removed and replace with other crystal. Just remove the crystal and put other crystal on turn pin without soldering.

**P** – Reserved for 2 x 16 LCD Display. User may solder 2x16 LCD display at this space if want to use it.

Figure below shown pin connection for 'Label P' (2x16 LCD Display).

Pin	Name	Pin function	Connection
1	GND	Ground	GND
2	VDD	Positive supply for LCD	5V
3	Con	Brightness adjust	Connected to a preset to adjust brightness
4	RB4	Select register, select instruction or data register	Pin RS of LCD
5	GND	Ground	GND
6	RB5	Start data read or write	Pin E of LCD
7	RD0	LCD Data bus pin	Pin D0 of LCD
8	RD1	LCD Data bus pin	Pin D1 of LCD
9	RD2	LCD Data bus pin	Pin D2 of LCD
10	RD3	LCD Data bus pin	Pin D3 of LCD
11	RD4	LCD Data bus pin	Pin D4 of LCD
12	RD5	LCD Data bus pin	Pin D5 of LCD
13	RD6	LCD Data bus pin	Pin D6 of LCD
14	RD7	LCD Data bus pin	Pin D7 of LCD
15	VDD	Backlight positive input	VDD
16	B/L	Backlight negative input	Connect to JP8

Figure below shown pin connection for 'Label O' (Turn pin for crystal).

Pin	Name	Pin function	Connection
RA6	OSC2	Crystal	Turn pin (JP7)
RA7	OSC1	Crystal	Turn pin (JP7)

Figure below shown pin connection for 'Label I' (Push button).

Pin	Name	Pin function	Connection
RB0	SW1	Digital Input	'SW1' SWITCH
RB1	SW2	Digital Input	'SW2' SWITCH