

**AUTOMATED GUIDED VEHICLE (AGV)
USING 68HC11 MICROCONTROLLER**

ZURIA ZAIDURA HASSAN

KOLEJ UNIVERSITI KEJURUTERAAN & TEKNOLOGI MALAYSIA

KOLEJ UNIVERSITI KEJURUTERAAN & TEKNOLOGI MALAYSIA

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **AUTOMATED GUIDED VEHICLE (AGV)**
USING 68HC11 MICROCONTROLLER

SESI PENGAJIAN: 2005/2006

Saya **ZURIA ZAIDURA BT HASSAN (831019-06-5398)**
(HURUF BESAR)

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

1. Tesis adalah hakmilik Kolej Universiti Kejuruteraan & Teknologi Malaysia.
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:

LOT 870 KM-8 JLN ENDAU-ROMPIN
KG JANGLAU 26820 KUALA ROMPIN
PAHANG

HAMDAN BIN DANIYAL
(Nama Penyelia)

Tarikh: **20 APRIL 2006**

Tarikh: : **20 APRIL 2006**

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).

“I hereby acknowledge that the scope and quality of this thesis is qualified for the
award of the Bachelor Degree of Electrical Engineering (Electronics)”

Signature : _____

Name : HAMDAN BIN DANIYAL

Date : 20 APRIL 2006

AUTOMATED GUIDED VEHICLE (AGV)
USING 68HC11 MICROCONTROLLER

ZURIA ZAIDURA HASSAN

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

Faculty of Electrical & Electronics Engineering
Kolej Universiti Kejuruteraan & Teknologi Malaysia

APRIL, 2006

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

Signature : _____

Author : ZURIA ZAIDURA BINTI HASSAN

Date : 20 APRIL 2006

*Dedicated to my beloved momma, dad, brothers, sisters,
nephew, niece, and to all Malaysians.*

ACKNOWLEDGEMENT

First and foremost, I am very grateful to the almighty Allah S.W.T for giving me this opportunity to accomplish my Final Year Project.

Personally, I would like to thank my supervisor, Mr. Hamdan bin Daniyal for his concern, guidance and full support in this project. Without his support and advice, this project will not be a success. Certainly, he is the best supervisor I ever had, who is willing to spend times to help me in any way he could. Secondly I would like to thank Mr. Abdul Halim bin Mohd Hanafi for his support, contribution of ideas and knowledge and also borrowing me some components during the testing of the circuit. Honestly, his support and kindness do impress me. Also not forgetting, my brother, Mr. Khairull Anuar bin Hassan for his willingness to spend his times helped me in buying some of the important parts and components, which were quite unreachable for me to get to the place. Fortunately he got the components for me in time, so letting me to get the components without any difficulties.

Thank you, to some of my course mates with their sharing and help during my difficulties completing my job of this project. Last but not least, I would like to thank to my family for their encouragement and all the people who in one way or another contributed to the success of this project.

Thank you very much. I do appreciate and remember everything from those who give their sincere help for life. I'm grateful to God and may God bless all of you.

ABSTRACT

Nowadays the creations of Automated Guided Vehicle (AGV) model can be found from all over the countries, as it give many advantages in our lives. It works just like a robot as it is able to sense and response to the environment. Considering that, AGVs should be well developed to optimize it's benefits to our own living. The aim of this project is to build a prototype of an Automated Guided Vehicle (AGV) model that can move on a flat surface with its two driving wheels and a free wheel. The prototype is able to follow line on floor with the M68HC11 microcontroller as it main brain that control all the navigation and responses to the environment. The ability to follow line on floor is an advantage of this prototype as it can be further developed to do more complicated task in real life. To follow the line, the microcontroller is attached to a sensor that continuously reflecting to the surface condition. Therefore, this project involves of designing and fabrication of the hardware and circuitry. The key study in this project is the algorithm designed in assembly language, embedded in the microcontroller.

ABSTRAK

Pada masa kini, penciptaan model AGV dapat dijumpai di setiap negara disebabkan kelebihan yang diperolehi daripadanya di dalam kehidupan kita. Model AGV dapat bekerja seperti sebuah robot yang boleh merasai dan bertindakbalas kepada persekitaran. Tujuan projek ini adalah untuk membina sebuah model AGV yang dapat bergerak diatas permukaan rata dengan menggunakan dua roda pandu dan satu roda yg bebas berputar sebanyak 360 darjah. Model AGV ini juga dapat bergerak mengikut garisan yang ditetapkan diatas permukaan rata dengan adanya pengawal M68HC11, yang bertindak sebagai pusat kawalan yang mengawal semua pergerakan dan tindakbalas kepada persekitaran. Kebolehan untuk bergerak mengikut garisan diatas permukaan lantai adalah satu kelebihan kepada model AGV ini yang seterusnya dapat diperkembangkan untuk melakukan kerja yang lebih rumit di masa akan datang. Untuk model ini bergerak megikut garisan, sebuah sensor digunakan dan disambung kepada litar pengawal. Oleh yang demikian, projek ini melibatkan penyambungan litar dan juga pembangunan program di dalam bahasa himpunan yang dimasukkan ke dalam litar pengawal.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF APPENDICES	xiii

CHAPTER I: INTRODUCTION

1.1	AGVs	1
1.2	Background	2
1.3	Problem Statement	2
1.4	Objective and Scope of The Project	3

CHAPTER II: LITERATURE REVIEW

2.1	Automated Guided Vehicle Built worldwide	5
2.1.1	Mobile Post Distribution System (MOPS)	5
2.1.2	ParkShuttle AGVs of Amsterdam's Schiphol Airport	7
2.2	Applications of Mobile Robots	8
2.3	Line Following Robot	10
2.3.1	Kerwin's Line Following Robot	11
2.3.2	Sandwich, The Line Following Robot	11

CHAPTER III: ITEM SELECTIONS

3.1	Real Robots	13
3.2	Hardware Development	20
3.2.1	DC Motors and The Gears	20
3.2.2	Motor Driver	22
3.2.3	Line Follower Sensor	28
3.2.4	68HC11 Microcontroller	31
3.2.5	68HC11 Internal Analog to Digital Converter	35
3.2.6	Power Supply Unit	38
3.3	Software Development	39
3.3.1	Communication between PC and 68HC11	39
3.3.2	Programming Tools	40

CHAPTER IV: DEVELOPED PROTOTYPE

4.1	Detail Block Diagram	42
4.2	Circuit Diagram	44
4.3	Algorithm	46
4.4	Final Prototype	47

4.5	List and Cost of The Components	50
 CHAPTER V: RESULTS AND DISCUSSIONS		
5.1	Early Result	51
5.1.1	Black Line Tracking	51
5.2	Implementing Realistic Application	55
 CHAPTER VI: CONCLUSION AND SUGGESTIONS		
6.1	Conclusion	59
6.2	Suggestion for Future Work	59
 REFERENCES		
REFERENCES		61
APPENDIX A		63
APPENDIX B		72
APPENDIX C		74
APPENDIX D		75
APPENDIX E		76

LIST OF TABLE

Table number	Title	Page
2.1	Applications of Mobile Robots	9
2.2	Line Following Method	10
3.1	Components and parts from issue 1	14
3.2	Components and parts from issue 2	16
3.3	Components and parts from issue 3	17
3.4	Components and parts from issue 4	18
3.5	Components and parts from issue 11	19
3.6	Components and parts from issue 12	20
3.7	Connector details of motor driver board	26
3.8	Truth table (a)	27
3.9	Truth table (b)	28
3.10	Aimed the sensor's units with hands	30
3.11	Aimed the sensor's units on various materials	30
3.12	Aimed the sensor's units on white surface with a drawn black line	31
3.13	Advantages of microcontroller M68HC11	33
3.14	Features of HClload	41
4.1	List and cost of components	50
5.1	Table of early result summary	55
5.2	Hexadecimal values from result registers	57

LIST OF FIGURE

Figure number	Title	Page
2.1	The mobile Post System (MOPS)	6
2.2	The ParkShuttle AGVs of Amsterdam's Schiphol Airport	7
2.3	The Kerwin's line following robot	11
2.4	Sandwich, the small and uncomplicated line following robot	11
3.1	Chassis	14
3.2	Tyres	14
3.3	Gear housing top section	15
3.4	Gear housing base section	15
3.5	Gear housing middle section	15
3.6	Wheel hubs with pre-assembled axle and gear	15
3.7	Spur gears	16
3.8	DC motor (Left motor)	16
3.9	DC motor (Right motor)	16
3.10	Front bumpers (2 sections)	17
3.11	Rear bumpers (2 sections)	17
3.12	Front and rear bumper rubber	18
3.13	assembled castor tyre, hub and axle	18
3.14	Castor housing	18
3.15	Motor power board	19
3.16	Line Follower Housing	19
3.17	Large 'O' ring	19
3.18	Line Follower sensor	20
3.19	Basic connection of H-Bridge circuit	22
3.20	NPN off	23
3.21	NPN on	23

Figure number	Title	Page
3.22	PNP off	24
3.23	PNP on	24
3.24	motor off	24
3.25	motor on	25
3.26	simplified motor on	25
3.27	Circuit of motor power board connected to power supply and motors.	27
3.28	The circuit of line follower sensor using two IR LEDs and two phototransistors.	29
3.29	Circuit of microcontroller in Bootstrap mode	34
3.31	Voltage Regulator	39
3.32	Interconnection in DB9	39
3.33	Interfacing PC communication port with microcontroller	40
4.1	Block diagram for the whole system	42
4.2	Circuit diagram for the whole system	44
4.3	Flow chart for black line tracking	46
4.4	Left side view	47
4.5	right side view	47
4.6	Top view	48
4.7	Bottom view	48
4.8	Front view	49
4.9	Rear view	49
5.1	Smooth curve of black line with line width = 1.8 cm	52
5.2	Rounded 90 degrees bend of black line with line width = 1.8cm	53
5.3	Sharp 90 degrees bend of black line with line width = 1.8 cm	53
5.4	Sharp 90 degrees bend of black line with line width = 2.6 cm	54
5.5	Sharp 90 degrees bend of black line with line width = 3.6 cm	54
5.6	Sharp 90 degrees bend of white line on cement floor with line width = 3.6cm	56
5.7	Rounded 90 degrees bend of white line on dark cement floor with line width = 3.6cm	56

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Technical Data of MC68HC11	63
B	Listing File and Source Code	72
C	ASM File	74
D	Basic Connection of Bootstrap Mode	75
E	CD Content	76

CHAPTER 1

INTRODUCTION

This chapter will discuss on the term AGVs. What is AGVs? Some brief history of AGVs and background are discussed. The problem statement stated the reasons to the creation and development of AGVs around the world. The objective and scope of this project is also presented in this chapter.

1.1 AGVs

There are many definitions of AGVs, different according to points of view.

Wikipedia, the free encyclopedia, defines AGVs as:

“A robot that been used highly in industrial applications to move materials from point to point”

The American Society of Safety Engineers (ASSE) defined AGVs as:

- a. Machines without drivers that can move along pre-programmed routes, or use sensory and navigation devices to find their own way around.
- b. Vehicles that are equipped with automatic guidance systems and are capable of following prescribed paths.
- c. Driverless vehicles that are programmed to follow a guide path.

1.2 Background

The creations of Automated Guided Vehicle (AGV) have been around since the 1950's and the technology was first developed by Barret Electronics from Grand Rapids, Michigan. It was then developed by the Europeans in the 1970's and nowadays AGVs can be found in any countries. One of the first AGVs was a towing vehicle that pulled a series of trailers between two points, and today's there are many task given to AGVs and they also have their own name and potentials.

Considering the full potentials and advantages of the Automated Guided Vehicle (AGV) in our livings, it is valuable to do this project, as it also will be the first step towards the creation of more intelligent technology or system. The simplest AGV model may use just a sensor to provide its navigation and can be the complex one with more sensors and advance systems to do the task. They can work or do the task everywhere needed but the safety for the AGV as well as the people and environment surround it must be provided.

The AGVs is just the same as mobile robot, which can moves from one place to another to do their task, but mostly the mobile robot is used for difficult task with dangerous environment such as bomb defusing. Furthermore, the mobile robot can be categorized into wheeled, tracked, or legged robot. Although the AGVs may not be glamorous of robots, but their work, which usually menial, are often be essential to the smooth running of factories, offices, hospitals, and even houses. They can work without any complaint around many workplaces all over the world.

1.3 Problem statement

There are many reasons which yield to the creation of Automated Guided Vehicle (AGV) around the world. Mostly the reason is to overcome the logistic problems that often occurred in the workplaces and to make improvement to the facilities provided in the workplaces. Usually the AGVs are implemented in factories, hospitals, offices, houses, and even can be found anywhere outdoors without the people surround realized it.

In the industries or factories, the AGVs can ease the physical strain on human workers by performing tiring tasks, such as lifting and carrying heavy materials, more efficiently with no signs of fatigue creeping in. They can carry far more than human workers, and their movements can be tracked electronically at all times. Their movements can be timed to feed or collect products or materials from the workcells in the factories.

Besides that, in the hospitals thousands of staff spends a portion of their day moving medical supplies, bedding, medicines and other equipment around large hospitals. By using the AGVs, the strain on the workers can be ease as well as the hospital's system would be more smart and systematic without any bad complaint from the patients and people. AGVs also capable of both cutting cost and releasing more staff hours to tend and care for patients.

Therefore it is very significant that the valuable knowledge on AGV construction is studied and be further implemented from the result of this project. It is due to its advantages to our own living and technology.

1.4 Objective and Scope of The Project

The objective and scope of this project is to create an AGV model that can follow a trail of line on a flat surface horizontally. This AGV model is using M68HC11 microcontroller to control all navigation during its operation. In other words, the microcontroller acts just like the brain for the model that controls all operation of the system.

The model is a three-wheeled mobile robot that has the ability to follow line on floor. There are three wheels including two driving wheels controlled by two motors and a free wheel in front that is able to rotate 360°. With three wheels, both driving wheels are always in contact with the surface, because of the robot's steering relies on both its driven wheels being in contact with the surface at all times.

This project consists of four main stages, which are theoretical design, mechanical fabrication, electronic hardware design and as well as algorithm design in assembly language. The matter to be considered is how the robot can follow the trail of line continuously. It is also important to choose the most suitable microcontroller, actuators, and sensors to achieve the project objectives.

CHAPTER 2

LITERATURE REVIEW

2.1 Automated Guided Vehicle Built worldwide

Some of the Automated Guided Vehicles (AGVs) that are well known are discussed in brief.

2.1.1 Mobile Post Distribution system (MOPS)

MoPS or Mobile Post Distribution System (Tschichold, Vestli, Schweitzer, 1999) is a research AGV developed at the Institute of Robotics in Zurich, Switzerland. It is used to transport mail around the Swiss Federal Institute of Technology in Zurich. MoPS is powered up by rechargeable batteries which give it a 4-hour active life, weighs around 90kg and can carry up to 50 kg of postal payload. It is also capable of hot-swapping its own batteries pack, thus ensuring 24h availability.

The MOPS provide services of picking up boxes with incoming mail at the ground floor of the five floor building, which is sorted by human first, delivering them to the secretaries' offices, subsequently bringing back the outgoing mail to the ground floor station. It is also capable of switching floors by sending an infrared signal to the building's lifts. As the building is open to the public, protection against theft of the mail is provided by motorized blinds over the pigeon-hole mail points, which can be opened by the robot and by authorized staff.

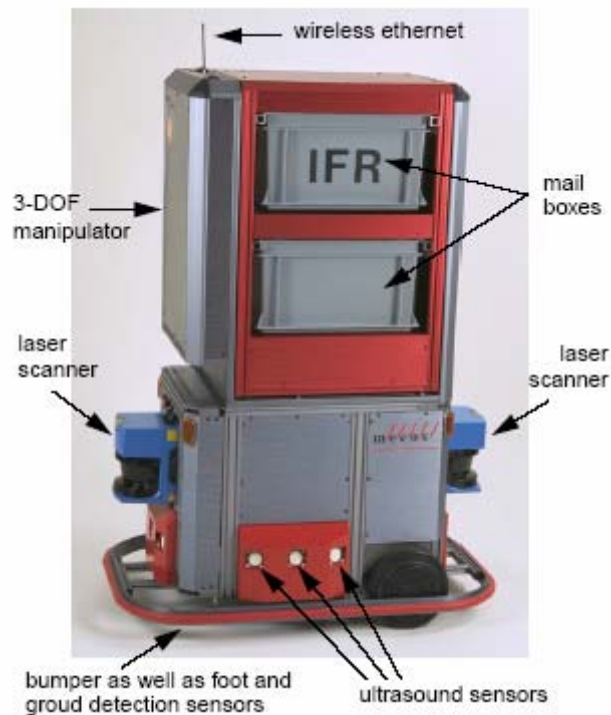


Figure 2.1 The Mobile Post System MOPS. The upper part of the system constitutes a 3-DOF manipulator / storage mechanism for two mail boxes. The lower part is a mobile robot platform with two drive and one castor wheels.

MOPS navigation is based on the recognition of natural landmarks which are compared to data of the building layout stored on the robots processor. The system has a highly accurate localization and position control system, which enables MOPS to dock onto the pigeon holes and to load/unload boxes of mail. A hybrid navigation scheme combining graph-searching with a situation based behavior selector and appropriate behaviors is used. The generated paths are close to being optimal, and the tolerance towards obstacles is high.

The interaction with humans takes place on various levels. The robot can be addressed by the secretaries and the maintenance offices by wireless internet communication, giving the robot its mail orders. The robot is supervised from the maintenance office, dealing with exceptional situations. Locally, service personnel can interact with the robot, having direct access to various robot functions.

The mobile robot has a distinctly visible status display, and humans crossing its way can interact with the machine which has to react to human given signals. The software is based on the object oriented real-time framework XOberon, fulfilling real-time requirements, and running on a PowerPC.

2.1.2 ParkShuttle AGVs of Amsterdam's Schiphol Airport



Figure 2.2 The ParkShuttle AGVs of Amsterdam's Schiphol Airport.

The ParkShuttle (FROG Navigation Systems) is an automatic navigating vehicle which provides transportation for passengers. It is a people mover system. There is no driver onboard, instead a computer and an electronic navigation system do the driving. This ParkShuttle has a safety system of sensitive and intelligent sensors. The sensors scan the area in front of the vehicle and will decelerate or stop the vehicle when an unknown obstacle is detected.

An additional safety feature is provided by the bumper system that brings the vehicle to an immediate halt when it is impressed. In addition, the vehicle has emergency stop buttons (both inside and outside) that can be operated by the passengers. The speed is limited to 40 km/h obtain a good ride quality.

The ParkShuttle vehicle runs on four rubber tires. Traction is provided by an electric motor powered by a rechargeable battery. Up to 100 km can be covered on one battery-load. It has a capacity of 10 passengers, 6 seated and 4 standees. It is easy to get into and out of the vehicle (wheelchair accessible) and provides good all-round visibility. Inside the vehicle is a console on which the passengers can indicate their destination.

Each vehicle is also fitted with an information display that announces the stop at which the vehicle has arrived. The maximum load is 800 kg. The maximum vehicle weight is monitored by means of weight sensor.

As soon as the total weight of the passengers and cargo exceeds the limit, the vehicle will refuse to depart and a message will automatically be announced. Sensors fitted in the doorways monitor the entrance and exit of passengers. The vehicle will never depart while a passenger is in the process of boarding and exiting.

People mover systems are often using some form of mechanical guidance, and these systems are already operating in different locations around the world. Vice versa this ParkShuttle is operating without mechanical guidance as it will find its way automatically traveling on a simple asphalt track with electronically guidance.

The main characteristics of this transportation system are small transportation units, high frequency, high density work, automatic operation, on-demand operation and simple infrastructure on ground level.

2.2 Applications of Mobile Robots

There are so many mobile robots can be seen around the countries over the world. Furthermore the applications of mobile robots can be seen in many fields. Mobile robots applications include in various fields (Alonzo Kelly, 1996) as listed in the Table 2.1.

Table 2.1 Applications of Mobile Robots

Application	Details
Medical service	<ul style="list-style-type: none"> a. Deliver food, water, newspapers, and linens. b. Medication, administrative reports hazardous material, biological waste.
Commercial cleaning	<ul style="list-style-type: none"> a. Airports, supermarkets, malls, and factories. b. Lots of floor treatments like wash, sweep, mop, scrub, buff, wax, polish, vacuum, strip, shampoo, trash pick-up. c. Other unpleasant jobs like washing bathrooms, windows, upholstery.
Hazardous and energy	<ul style="list-style-type: none"> a. Bomb and mine mapping, retrieval and disposal. b. Nuclear plant inspection, steam generators. c. Hazardous waste storage tank inspection. d. Pipeline inspection crawlers. e. High tension power line inspection.
Space	<ul style="list-style-type: none"> a. Terrestrial inspections of space vehicles. b. Satellite on-orbit inspection. c. Planetary exploration.
Undersea	<ul style="list-style-type: none"> a. Drilling platform inspection. b. Transatlantic cable installation and maintenance. c. Exploration.
Material Handling	<ul style="list-style-type: none"> a. AGVs operate successfully today in highly structured automotive and electronics factories. b. Loading and unloading of trucks, trains, ships, and planes.
Civil Transport	<ul style="list-style-type: none"> a. Aircraft inspection. b. Automated and/or intelligent highway vehicles.
Personal	<ul style="list-style-type: none"> a. Assistants for handicapped and elderly individuals. b. Assist with personal hygiene, working at home, recreation. c. Seeing Eye robot, smart self-navigating wheelchair.

2.3 Line Following Robot

Line following robot is generally a wheeled mobile robot. The method of line following varied depending on the number of sensors available and the type of line to be followed. There are four methods identified including edge following, line search, line trap, and cross-over. These four methods are different in number of sensors that used and also the results that will be obtained are different. With only one light sensor, the robot will have to know where the line is, or spends time searching to find it. Whereas with two light sensors, the robot is possible to remember which direction the line went. With more sensors, the result that will be obtained would be more excellent and the robot will be more intelligent. The line following method can be listed in Table 2.2 below.

Table 2.2 Line Following Method

Method	Characteristic
Edge following	Stay on the edge of the line
Line search	Stay on the line
Line trap	Keep the line between sensors
Cross-over	Move back and forth over the line

For each method, there are different characteristics and give choices to the builder of the line following robot to use which one from the list. The choice is depend on the function of the robot to be build as each method has its own advantages and disadvantages.

There are many line following robot have been built worldwide. Some of them will be discussed in brief.

2.3.1 Kerwin's Line Following Robot

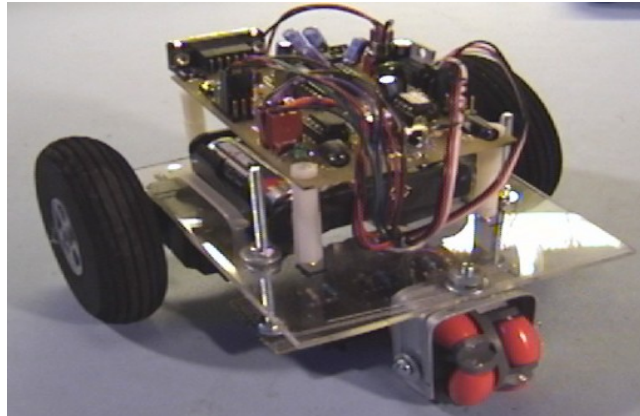


Figure 2.3 The Kerwin's line following robot using three matched IR transmit/receive pairs.

The Kerwin's line following robot (ranchbots) is a design with Futaba S-148 servo motors mounted to the bottom of the plexiglass. It has three wheels with the front wheel is the omni-directional wheel. The sensor system consists of an array of three matched IR transmit/receive pairs mounted on a circuit board that can be raised or lowered to fine tune the sensitivity. It uses the Atmel microcontroller as the controller part. The microcontroller takes input from sensor array and drives the servo motors in response.

2.3.2 Sandwich, The Line Following Robot



Figure 2.4 Sandwich, the small and uncomplicated line following robot.

Sandwich, the line following robot (Robot Room) is designed with smaller and lighter body, motors, connectors, and comparator as the brain. It does not use the microcontroller, thus no programming are developed. It uses two pairs of cadmium-sulfide photoresistors as the sensors instead of IR LEDs.

Sandwich is able to follow either light or dark paths, although the driving isn't as smooth and can't take as sharp of turns. It won't stop when the path ends, and can't automatically detect if the line is light or dark, thus manually detect the line using switch.

CHAPTER 3

ITEM SELECTIONS

In developing this expected prototype, it is important to choose the best materials or parts and components by considering the factors that contribute to determine the correct materials. The factors are such as the environment in which this prototype works, the reliability, cost effective, and any other factors that influence in which material choice is the correct one. By choosing and using the wrong components will lead to much more problems to the project's development, causing damage to the other parts and also escalating cost. Generally, the components and parts for building a robot is costly and therefore avoiding of wrong pick up materials is a suggestion to all the robot builder beginners.

3.1 Real Robots

In this project, it is decided to use the mechanical parts from the Real Robots. The components are good enough, reliable, and cost effective as compared to other components choices. For this project, it is not all of the Real Robot's part been used, as just the issues 1, 2, 3, 4, 11 and 12 of the Real Robots Magazines are ordered and the component for each issues are listed in Tables below.

Table 3.1 Components and parts from Issue 1

Issue 1
<ul style="list-style-type: none"> a. Chassis. b. Gear housing (base section x 2, middle section x 2, and top section x 2). c. 4 short self-tapping screws (to fix base section to chassis). d. 4 long screws (to fix housings together). e. 2 tyres (for rear wheels). f. 2 wheel hubs with pre-assembled axle and gear. g. 4 x spur gears. h. 4 x gear shafts.

**Figure 3.1** Chassis**Figure 3.2** tyres



Figure 3.3 Gear housing top section

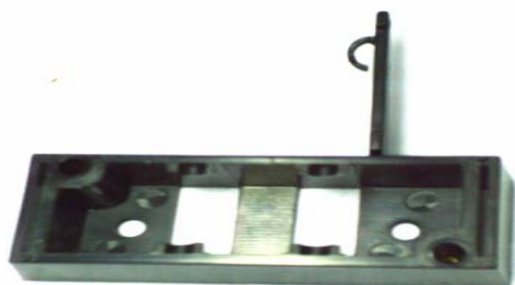


Figure 3.4 Gear housing base section

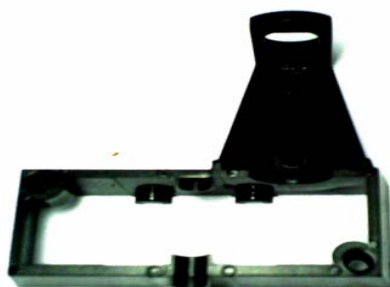


Figure 3.5 Gear housing middle section



Figure 3.6 wheel hubs with pre-assembled axle and gear

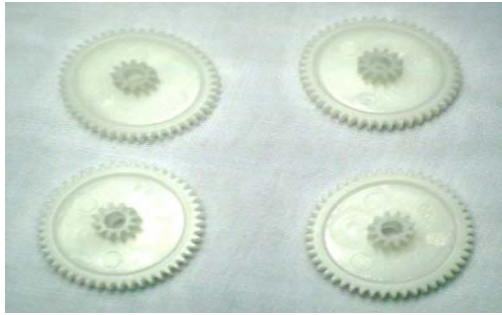


Figure 3.7 spur gears

Table 3.2 Components and parts from Issue 2

Issue 2	
a.	2 motors.
b.	Battery box (6V).
c.	Battery box lid.
d.	4 self-tapping screws.
e.	Sachet of grease.



Figure 3.8 DC motor (Left motor)



Figure 3.9 DC motor (Right motor)

Table 3.3 Components and parts from Issue 3

Issue 3	
a.	Castor housing.
b.	Castor hub.
c.	Castor tyre.
d.	Castor axle.
e.	Switch (with wires and connectors attached).
f.	2 small self-tapping screws (to fix switch).
g.	Front bumper (2 sections).
h.	Front bumper rubber.
i.	Rear bumper (2 sections).
j.	Rear bumper rubber.
k.	6 self-tapping screws.

**Figure 3.10** Front bumpers (2 sections)**Figure 3.11** Rear bumpers (2 sections)

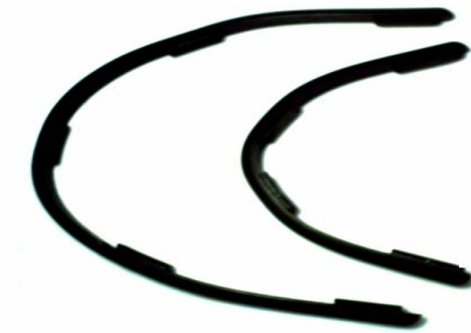


Figure 3.12 Front and rear bumper rubber



Figure 3.13 assembled castor tyre, hub and axle



Figure 3.14 Castor housing

Table 3.4 Components and parts from Issue 4

Issue 4	
a.	Motor power board.
b.	Screws (x2).
c.	Motor tester board.
d.	Screws with nuts (x2).

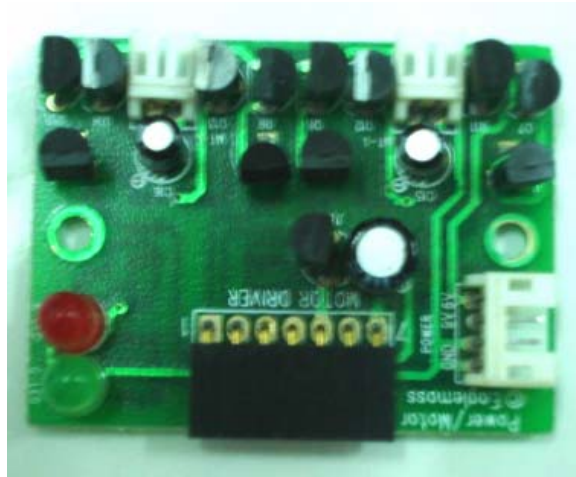


Figure 3.15 Motor power board

Table 3.5 Components and parts from Issue 11

Issue 11
<ul style="list-style-type: none"> a. Line follower housing. b. Large 'O' ring. c. 2 self-tapping screws.



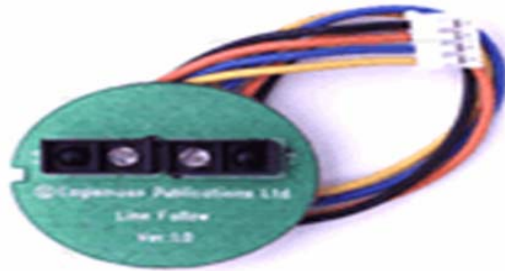
Figure 3.16 Line Follower Housing



Figure 3.17 Large 'O' ring

Table 3.6 Components and parts from Issue 12

Issue 12
a. Line follower sensor with wires.

**Figure 3.18** Line Follower sensor

3.2 Hardware Development

The hardware development consists of the process of developing the components and parts to be assembled and at last become the final product of the prototype. It is the important part of the project that determines the successful of the project.

3.2.1 DC Motors and The Gears

The selection of motors consists of DC motors, stepper motors, and servo motors. It is important to consider on the strength and weakness of each of the motors. The type of motors to be used in this project is based on its intended usage and other factors such as cost, availability and physical characteristic of the motor such as dimensions and weight that must be considered thoroughly before making any decision on the motor used.

In this project, DC motors are much preferred rather than the other types because of some factors such as the prototype need to have continuous mechanical movement along the line. A small DC motor would be favorable to be use since it is very reliable,

very strong for its size, and very inexpensive. It can also be made to run on low voltages, have good electrical to mechanical conversion coefficients and is available in an immense variety of sizes and formats. Thus in this project with just a few volts been used, the small DC motors would be the best choice. The DC motors from the Real Robots is the small DC motors that best suited to be used in this project.

There are two DC motors used for this prototype, one for each RIGHT and LEFT driving wheels. Each of the motors carries a 12-tooth gear that will complete the gear train for each side of the prototype. The motors are powered by four 1.5V AA batteries, which give a total voltage of 6V.

The details of the operation of the DC motors will not be discussed, but in brief the DC motors converts electrical energy into mechanical movement. This is due to inside the motors, there are copper wires wrapped tightly around an iron core (which is the rotor) and surrounding this, is a permanent magnet (the stator). By switching on the motors, electric current will flow through the wires and around the core, thus causing it to become magnetized. The rotor that turns as it tries to align its own poles with those of the stator. Every half a turn, a vital component called a commutator reverses the direction of the current on the rotor. In other words, it swapping its poles and so keeping the motor turning.

A series of gears are connected to the motors to the wheels. This is important as to slow down the spindles of the motors that are extremely fast. Therefore, it is the job of the gears to step down the speed of the motor. In brief theoretically, each pair of the spur gears consists of little (12-tooth) and large (48-tooth) gears can steps the speed down by a factor of one quarter ($12/48$). For each side of this prototype motors, there are a total of 3 pairs of gears attached, so the whole gear train steps the speed down by $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} = 1/64$. Therefore for every 64 turns of the motor, the wheel turns only once.

Thus, the selection of the DC motors and gears must be absolutely correct to its intended usage, as to expect the optimize mechanical movement of the driving wheels.

3.2.2 Motor Driver

The motor driver is used to drive the motors of the robot. Since even a small motor requires high current to operate, the high current could certainly causing damage to the microcontroller if the motor is connected directly to the microcontroller.

Then, normally the motor driver or a relay is required to prevent the reverse current and protect the microcontroller chip. The resistor $10k\Omega$ of the motor driver circuit is also used to limit the rush current. The microcontroller is lacks of sufficient current to drive the motors, so the driver is needed to energize the stator of the motor.

The drivers for DC motors are H-Bridge circuit as well as L293B/L298N. For this project, the H-Bridge driver circuit is the best solution to be interface with the dc motors. Basically the operation of the H-Bridge circuit is as follow:

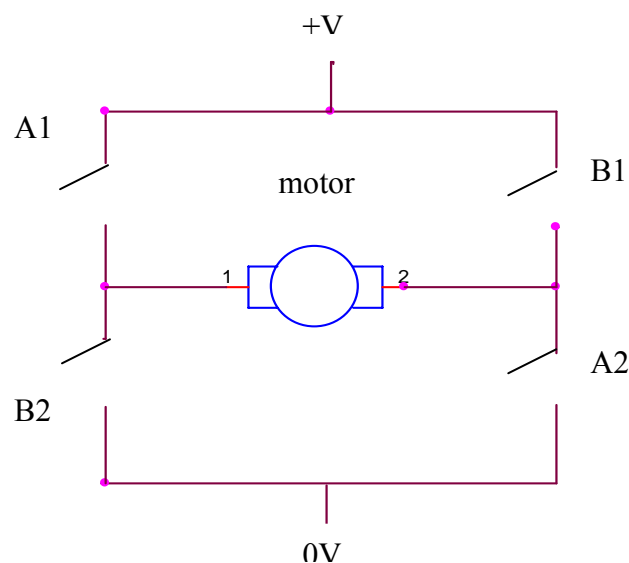


Figure 3.19 Basic connection of H-Bridge circuit

The circuit of Figure 3.19 consists of a motor, 2 supply rails and 4 switches. Any of the switches can be close at any time, giving the total of 16 different combinations, but only 3 of the combinations are useful. If the A1 and A2 switches are turn on, the current will flow in one direction through the motor, causing the motor to turn on. If the B1 and

B2 switches are turn on, the current will flow in the opposite direction through the motor, causing it to spin in the opposite direction. If all the switches are turn off, then there is no current will flow through the motor, and it is free to spin. The simplicity of this type of H-Bridge circuit makes it ideal for robotics.

The implementation of the H-Bridge motor driver can be done according to the circuit diagram. In this project, it is decided to use the Real Robots motor driver board considering the factor of reliability and size. The circuit of the motor driver board consists of two simple H-Bridge drivers, one for each motor. The 5 volt regulator is used for feeding power from the 9 volt battery to the microcontroller MC68HC11A1. The two LED's – the green one shows power is turned on (from the 9 volt battery), and the red one lights on when the 9 volt battery is getting low. In the H-Bridge connection, all the transistors are used as simple switches. The Figures 3.20 to Figure 3.26 are the explanation of the operation of the H-Bridge motor driver board in more details.

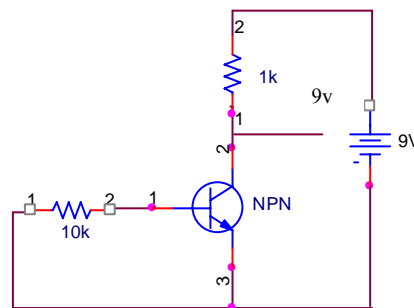


Figure 3.20 NPN off

The base has the same potential as (or slightly below) the emitter will cause the transistor to switch off.

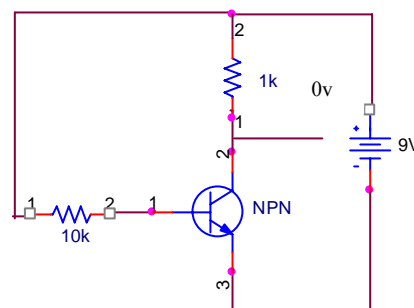


Figure 3.21 NPN on

Taking the base 'positive' of the emitter will cause the transistor to switch on.

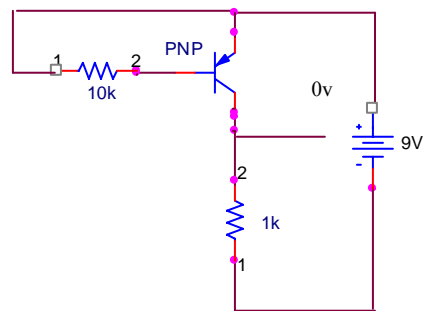


Figure 3.22 PNP off

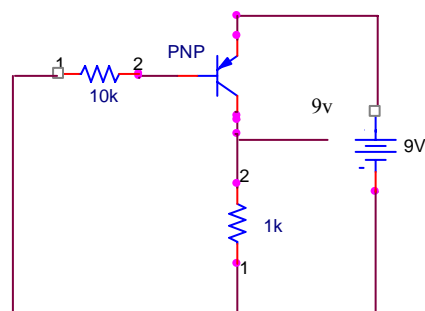


Figure 3.23 PNP on

Taking the base 'negative' of the emitter will cause the transistor to switch on.

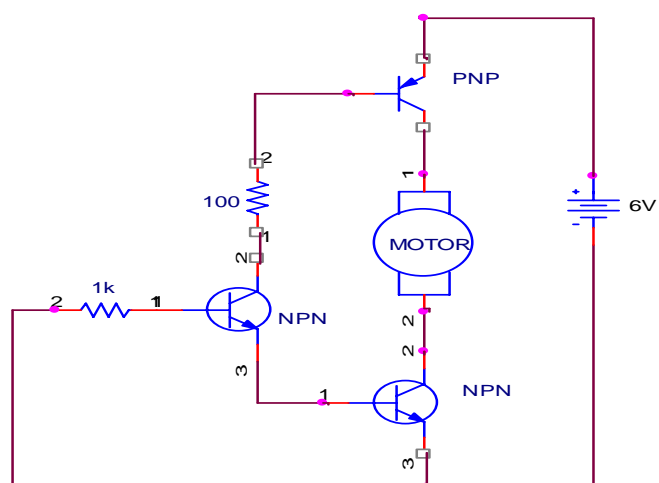


Figure 3.24 motor off

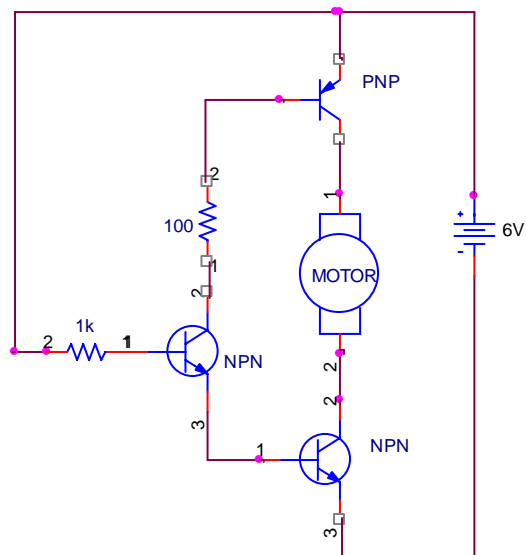


Figure 3.25 motor on

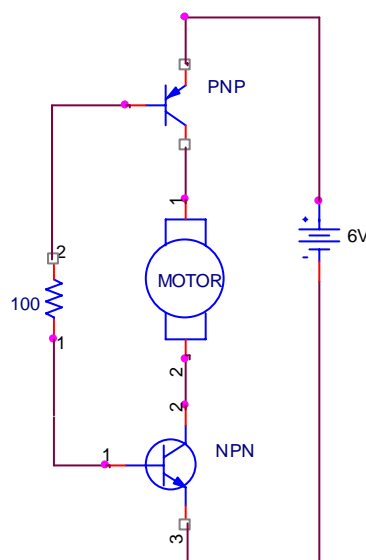


Figure 3.26 simplified motor on

The NPN is turned on, effectively connecting the 100ohm resistor directly between the bases of the two transistors. The 100 ohm resistor connects the base of the bottom (NPN) transistor towards the positive rail, thus turning it on, at the same time it connects the base of the upper (PNP) transistor towards the negative rail, thus turning that one on as well. In this way both transistors are turned hard on, and the motor has power and rotates.

If the NPN is turned off, the 100ohm resistor effectively disappears, so neither transistor is turned on, and the motor doesn't rotate. Each motor requires 2 sets of this circuit, one to

rotate clockwise, and the other to rotate anti-clockwise, by selecting one or other of the two inputs the motor is connected either way around.

3.2.2.1 Calibration of Motor Driver Board

The kind of experiment that has been accomplished during this calibration is to verify the 7-pin connector, 4-pin connector, the MT1 and MT2 of the motor driver board. In the beginning, the dc motors were attached to the respected MT1 and MT2. The Left motor with black wire is connected to MT1 and the Right motor with grey wire is connected to MT2 according to the reference data. It is important to recognize the correct connection between the motors and the motor driver board to avoid any wrong connection and troubles. From the reference data of the motor driver board, the connector details of the motor driver board are shown in the Table 3.7.

Table 3.7 Connector details of motor driver board

7 pin socket	4 pin socket	MT1	MT2
1- Right motor forward	1- GND 2- GND 3- 9 volts in 4- 6 volts in	LEFT DC motor	RIGHT DC motor
2- Right motor backward			
3- Left motor forward			
4- Left motor backward			
5- 5 volts to uC			
6- 6 volts			
7- GND			

During the experiment, the pin-6 of the 7-pin connector is not been used or not been connected with any connection. This is because, the 5 volts output voltage from the motor driver is a sufficient voltage value to drive the motors. At this stage, it is decided that the pin-6 will also not be used in future connection of the line following robot.

The motor driver supply the 5V to the microcontroller (uC), but at this stage of testing with motors, the 5V is the supply voltage for pin 1, pin 2, pin 3, and pin 4 of the 7-pin socket. The motor driver board as well as the motors are not yet been interfaced with the microcontroller. This is just to verify the connection and that the motors are moving as

it should be. The Figure 3.27 shows connection between the motor driver board, motors and power supply. The test procedure objective is to prove the truth table (a) of Table 3.8.

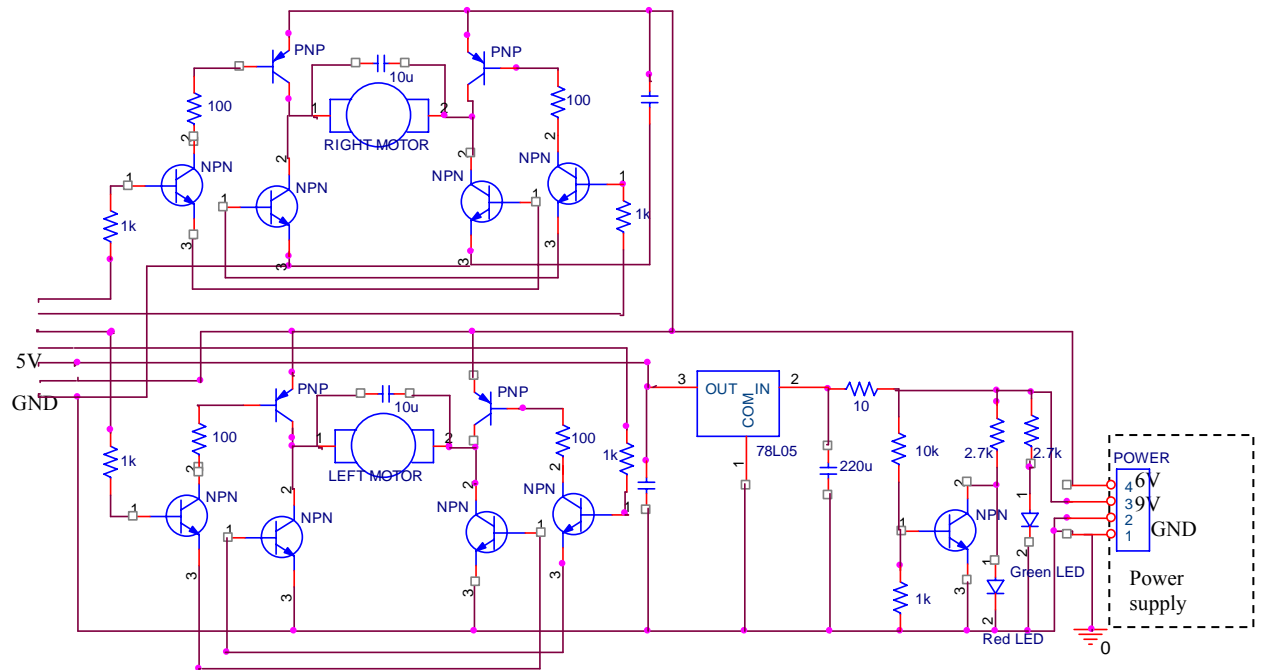


Figure 3.27 Circuit of motor power board connected to power supply and motors.

Table 3.8 Truth table (a)

Motor turn:	Pin 7	Pin 6	Pin 5	Pin 4	Pin 3	Pin 2	Pin 1
Forward	0		1	0	1	0	1
Backward	0		1	1	0	1	0
Right	0		1	0	1	1	0
				0	1	0	0
				0	0	1	0
Left	0		1	1	0	0	1
				0	0	0	1
				1	0	0	0

Logic “1” = 5V

Logic “0” = 0V/ GND

The truth table above is proved, and the driving wheels are turning according to the data in the table above. Therefore the truth table can be simplified as the truth table (b) shown by Table 3.9.

Table 3.9 Truth table (b)

Motor turn:	Pin 4 (PB0)	Pin 3 (PB1)	Pin 2 (PB2)	Pin 1 (PB3)	hex
Forward	0	1	0	1	\$0A
Backward	1	0	1	0	\$05
Right	0	1	1	0	\$06
	0	1	0	0	\$02
	0	0	1	0	\$04
Left	1	0	0	1	\$09
	0	0	0	1	\$08
	1	0	0	0	\$01

From the truth table (b), a simple program is constructed to see the robot moving forwards, backwards, turning right and left is been done. The motor driver is connected to the microcontroller to see whether the microcontroller can interface with the motor driver and the DC motors. From the Table 3.9, a value of hexadecimal = \$0A is store at port B of the microcontroller MC68HC11A1 and simply the driving wheels will move forwards. The truth table (b) is also proved as the driving wheels are moving according to listed hexadecimal values.

3.2.3 Line Follower Sensor

The Line Follower sensor uses a pair of IR sensors that aimed downwards close to the floor. Each pair consists of an IR LED and an adjacent photodiode (built as a single unit), so the light from the IR LED is reflected back by a light colored surface, and

absorbed by a dark colored surface and the photodiode is responding to the amount of light reflected back.

There are 4 pins through the connector of the sensor. From the reference data, it is identified that pin 1 (black wire) is for GROUND, pin 4 (yellow wire) is for Vcc, 5V. The circuit of the line follower sensor is shown in Figure 3.28.

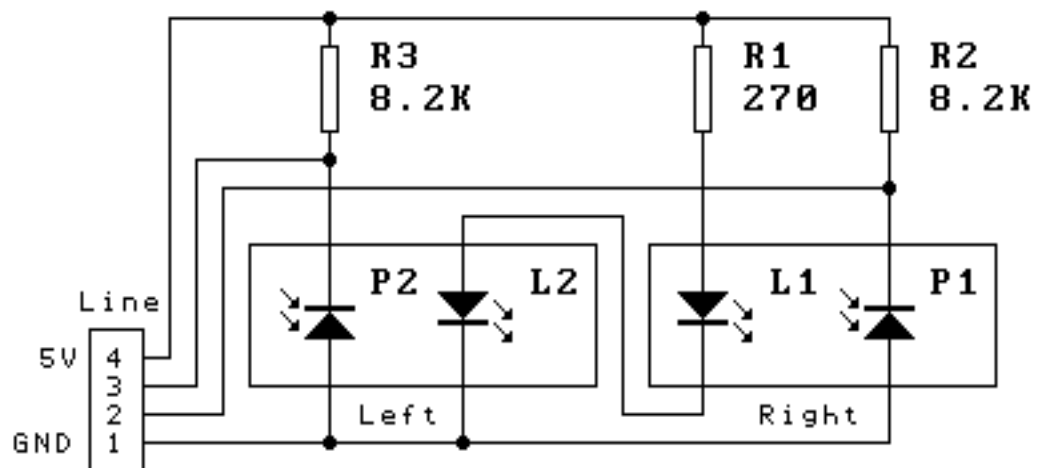


Figure 3.28 The circuit of line follower sensor using two IR LEDs and two phototransistors.

3.2.3.1 Calibration of Line Follower Sensor

The calibration for Line Follower sensor is to identify which one of the pin 2 and pin 3 of the Line follower sensor that functioning as Right or Left. During the testing, Channel 1 (CH1) of oscilloscope is connected to pin 2 (orange wire) and channel 2 (CH2) of oscilloscope is connected to pin 3 (blue wire). Pin 1 and pin 4 is connected to GND and 5V of power supply respectively.

At first, we close both the Right and Left units with hands. The oscilloscope shows the voltage at both channels, showing that the sensor's units can convert the physical quantity into an electrical quantity, which is the voltage. The readings show in the Table 3.10.

Table 3.10 Aimed the sensor's units with hands

	CH1	CH2
RIGHT	4.8V	1.8V
LEFT	1.8V	4.8V

It is shows that when we close the right unit with hand, the voltage on CH1 is greater than CH2 voltage. Vice versa, when we close the left unit with hand, the voltage on CH2 is greater than the CH1 voltage.

Then we move to test whether CH1 or CH2 are functioning as Right or Left. It is decided that CH2 is functioning as RIGHT and CH1 is functioning as LEFT. This is because, when testing by closing the unit of IR LED and photodiode by hand; CH2 is showing various voltage values while CH1 is showing constant value of voltage.

Table 3.11 Aimed the sensor's units on various materials

Situation	Voltage
Aimed the sensor's units downwards close to white surface of a piece of paper.	0.33V
Aimed the sensor's units downwards close to black surface (black ink marker)	3.5V
Aimed the sensor's units downwards close to black surface (black plastic material)	4.1V

The results from Table 3.11 show that both the Right and Left units give the same voltage readings, meaning that both the units are aimed downwards close to the same surface. Then, a more appropriate test is being made by aiming the sensor's unit on a piece of white paper with a bend of black line is drawn on it. The black line is made by the black ink marker. The results of the voltage are shown in the Table 3.12.

Table 3.12 Aimed the sensor's units on white surface with a drawn black line

Situation	Right	Left
The Right unit on white surface while Left unit on black line	0.375V	3.25V
Both the Right and Left units on white surface	0.33V	0.33V
Both the Right and Left units on black line	3.3V	3.3V
Left unit on white surface and Right unit on black line	3.61V	0.329V

It can be summarized that, if the sensor's unit are aimed downwards close to a bright or white surface the voltage reading is almost approximately zero, but if the sensor's unit are aimed downwards close to dark or black surface the voltage reading is almost approximately 5V.

3.2.4 68HC11 Microcontroller

Generally, there are numerous control methods to control a robot. The methods are ranged from classical method like PI controller, PD controller and PID controller to artificial intelligence such as Fuzzy Logic, Genetic Algorithm and Neural Networks. Microprocessor and microcontroller are also can be use to control robots which acts just like the brain for the robot.

The brain that controls this prototype is the Motorola MC68HC11A1 48 pins DIP microcontroller. The MC68HC11A1 can control the entire DC motors and the behavior of the line-following sensor. The MC68HC11A1 is in the bootstrap mode to allow serial communication interface. It allows programs to be assemble in Motorola S19 code to be loaded into the controller's internal RAM or EEPROM. The microcontroller can communicates with the personal computer (PC) through the serial communication port using the RS232 communication protocols.

Normally there are five ports in the microcontroller namely port A, port B, port C, port D, port E and port F. But for MC68HC11A1 there are just port A, port B, port C, port D and port E available and not all the ports are interfaced with application circuit. There are just 4 of port E are available for this 48 pins MC68HC11A1 (PE0, PE1, PE2, and PE3). The DC motors are interfaced with port B and the line following sensor is interfaced with port E.

Microcontroller also can be recognized as computer-on-a-chip optimized to control electronic devices which consist of CPU, ROM, RAM, I/O port and timer embedded together inside the single chip. It can follows instructions, read information, communicates, measures time and switches things on and off and also does other things too depending on the model. It is often criteria to be considered to which microcontroller to be used, such as the microcontroller should meet the computing needs for the task at cost efficiency and also has simple architecture and language.

A typical microcontroller contains all the memory and I/O interfaces needed, whereas a general purpose microprocessor requires additional chips to provide these necessary functions. Microcontroller is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor, the kind used in a Personal Computer (PC).

A microcontroller differs from a general-purpose CPU chip because the former generally is quite easy to make into a working computer, with a minimum of external support chips. The idea is that the microcontroller will be placed in the device to control, hooked up to power and any information it needs, and that's that.

Vice versa, a microprocessor requires all of these tasks to be handled by other chips. For example, some number of RAM memory chips must be added. The amount of memory provided is more flexible in the traditional approach, but at least a few external memory chips must be provided, and additionally requires that many connections must be made to pass the data back and forth to them.

Major microcontroller producers are Motorola which produces the M68HC11 and M68HC12. M68HC11 microcontroller is an 8 bit microcontroller which comes in 48pins, 52pins, 56pins and 64pins DIP, PLCC, CLCC, TQFP, SDIP and QFP packaging.

For this project, M68HC11 microcontroller is much preferred than other type of microcontroller because of some factors listed in the Table 3.13.

Table 3.13 Advantages of microcontroller M68HC11

Advantages
<ul style="list-style-type: none"> a. Offers various subsystems such as ADC, interrupts, and timers. b. Simple assembly language because the processor uses the von Neumann architecture. c. Meet the computing needs with various speed, packaging, and power consumption; various RAM and ROM sizes; multiples number of I/O pins. d. Various free sources of software and references: assembler, debugger, c-compiler, emulator, books, website. e. One of the popular and are often used in robotic and automotive and are closely associated with 6800 family. f. Economic in term of design as it requires MAX232 and serial port of pc.

3.2.4.1 Bootstrap Mode

The microcontroller is in the bootstrap mode, with minimum uses of external connections. To get the circuit of the microcontroller in the bootstrap mode, the MODA and MODB pins is connected to GND, as the mode of microcontroller's operation is determined by the status of MODA and MODB pins during the RESET operation. Since both the MODA and MODB = 0, the small on-chip at address \$BF40-\$BFFF is enabled when the 68HC11 is reset in the special bootstrap mode. The reset vector (\$BF40) at address \$BFFE and \$BFFF is loaded in the program counter and the 68HC11 proceeds to

3.2.5 68HC11 internal Analog to Digital Converter

The Line Follower sensor detects its environment and receives the signal of physical variables such as heat, temperature, and pressure which are analog in nature. The analog signal is the voltage that varies according to the environment it reflects. This analog signal can be process by the CPU as if it is first be converted to a digital value. Therefore, an analog to digital converter (ADC) is needed to perform this task. Generally, an analog to digital converter is a device that converts continuous signals to discrete digital numbers. It is also be defined as an electronic device that converts a voltage to a binary digital number.

The 68HC11 microcontroller has the built-in Analog-to-Digital (A/D) Converter subsystem to perform the conversion. The two basic specifications for the A/D Converter are analog input range and resolution. The input range for M68HC11 microcontroller is 0 to 5.12V. The analog interface must scale the transducer signal to fit within this range. For the Line Follower sensor, it is an advantage as this sensor is already operating in the 0 to 5V range. The resolution for 68HC11 is defined as:

$$\text{Resolution} = \frac{\text{Analog input range}}{2^n} \quad ; \text{ where } n \text{ is the number of bits}$$

The equation of resolution shows and gives a measure of accuracy. For the 68HC11, the resolution is 8 bits. It means that the ADC has:

- a. 8 digital outputs
- b. 2^8 or 256 possible binary output values from 00000000 to 11111111

Therefore for the 8 bit 68HC11 with input ranging from 0 to 5.12V, the resolution is:

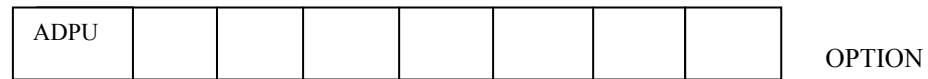
$$\text{Resolution} = \frac{5.12\text{V}}{2^8} = \frac{20\text{mV}}{1 \text{ bit}}$$

Meaning that the input must change by 20mV to change the digital output code by 1 bit.

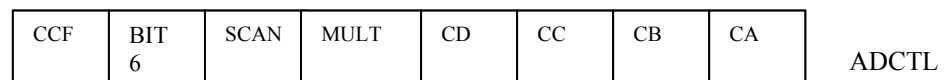
For the MC68HC11A1 48-pin DIP, there are four channels available for ADC, at pin-17, pin-18, pin-19, and pin-20. All the 4 pins are also the pins for port E (input port). Therefore, to let the port to be use as ADC, initialization of Port E (at address register

\$100A) as Analog-to-Digital Converter is the foremost of all steps. Without Initialization, the address register of \$100A will be recognized as input port by the microcontroller. The steps of initializations are as follows:

- a. Access the OPTION register, and set the ADPU bit to power up the ADC.



- b. Writing to control-status register (ADCTL) to initialize the unit and to begin conversion.



- c. Gives delay at least 100us
- d. Access the data from the result register; ADR1, ADR2, ADR3, and ADR4

3.2.5.1 Measurement of Voltage of Potentiometer at Port E1 (PE1)

This part is to test a known dc voltage and use the microcontroller's analog-to-digital converter to read its value, thereby learning how the ADC stores its result and how data can be read by the CPU.

In this measurement, a 10kΩ potentiometer is used and connected to port E1 at pin-18 of the MC68HC11A1. The microcontroller circuit is still in the bootstrap mode circuit interfaced with the potentiometer with the supply voltage of 5V supplied by the power supply unit. The connection of the microcontroller and the potentiometer is shown by Figure 3.30 below.

To take the measurement, the circuit is communicating with PC via the DB9 connection. The software JBug is used execute a simple program to initialize the microcontroller's Analog-to-Digital Converter program thus reading the voltage value when the resistance values of the potentiometer is varied.

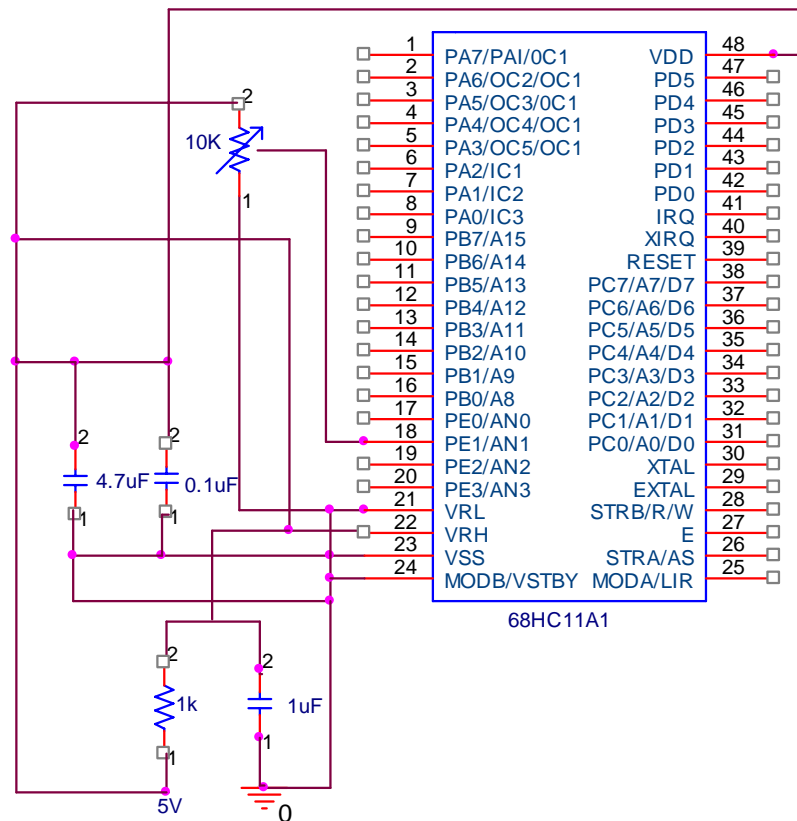


Figure 3.30 setting up a known dc voltage at Port E1 to the microcontroller in Bootstrap mode.

The result shown by the ADR2 result register can be traced by the JBug software. From the result, It shows that the calculated voltage value for a certain resistance value is not exactly the same with the measured value that is traced with the JBug software. The voltage value in hexadecimal has same difference between the calculated and the measured one, but the difference is not so much, just approximately differentiated at few mili volts (mV).

For the upcoming measurement of voltage values taking from the line follower sensor, the voltage range that is traced from the JBug software will be the reference value to make the programming for this project.

3.2.5.2 Measurement of Voltage from Line Follower Sensor

Since the measurement of potentiometer's voltage values is successful, it is a step forward to interface the line follower sensor to the microcontroller MC68HC11A1 that is still in the bootstrap mode circuit connection.

In this measurement, two of Port E are being used; Port E1 and Port E2. Thus a multi channel microcontroller's analog-to-digital converter is used. The initialization step of the ADC is still the same, but the value in hexadecimal stored in the ADCTL register is change to \$30.

Communicating with the JBug software, the line follower sensor also gives the reading of voltage values when the sensor is aimed on any surface that is dark and bright.

3.2.6 Power Supply Unit

The microcontroller is powered up by a low voltage value ranging from 0V to the maximum 5V. It will cause to the internal circuitry damage to the microcontroller 68HC11 if it is directly connected to a high value of supply voltage.

In this project, the voltage regulator is very important to turn the 9V supply from the battery to a constant 5V, to power on the microcontroller. This regulator is already interfaced on the motor driver board that is a smart enough connection.

The 6V of voltage value supplied by four of 1.5V batteries is just to power on the dc motors and this 6V is not connected directly to the microcontroller. The basic connection of voltage regulator to turn any input voltage to a constant 5V value is as shown by Figure 3.31 below.

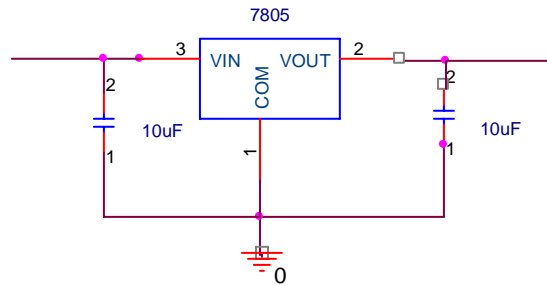


Figure 3.31 Voltage Regulator

3.3 Software Development

The software development includes the process of determining which communication port of PC to be interface with the microcontroller as well as the programming tools to be use. It is also the important part towards the completion of this project.

3.3.1 Communication between PC and 68HC11

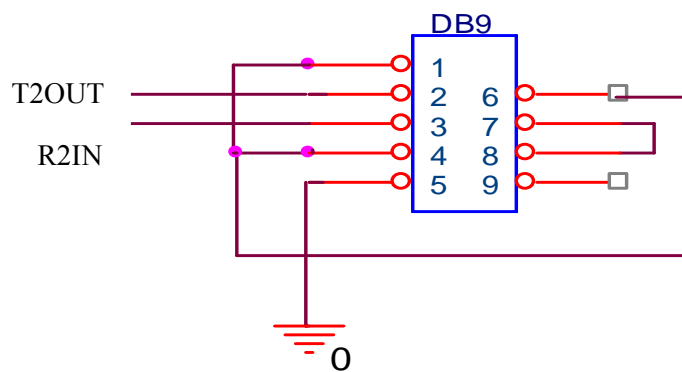


Figure 3.32 Interconnection in DB9

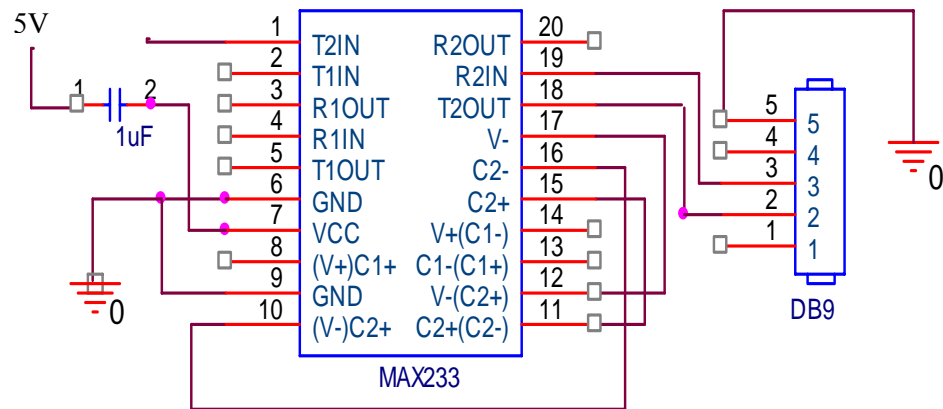


Figure 3.33 Interfacing PC communication port with microcontroller

3.3.2 Programming Tools

In this project, to JBug11 software and Hcload software are used to put the program into the microcontroller. This software can be used to load and erase the program as desired.

3.3.2.1 Hcload Software

Hcload is a Windows boot loader for many flavor of HC11. It is a solution for a powerful tool to program, erase, verify the EEprom, Eprom, Rom of a HC11. The Features in Hcload are shown in Table 3.14.

Table 3.14 Features of HLoad.

Features of HLoad
<ul style="list-style-type: none"> a. Check the communication between HLoad and the PC b. Program/Verify/Compare/Erase the EEprom of the HC11 c. Program/Verify/Compare the Eprom of the HC11 d. Program the config register of the HC11 e. Download/Execute code in external RAM f. Upload any internal/external memory range to PC in a S19 file g. Program/Verify/Compare an external EEprom h. Connect/Disconnect HLoad to CommPort

3.3.2.2 JBug Software

JBug is a real time debugger that allows the execution of program and to check memory and registers. It also allows programming of the EEPROM. It consumes approximately 256 bytes of RAM (\$0000- \$00FF) for the serial communication program (Talker). When it comes to debug Assembly program, it is make sure that the space for variables is not in used.

CHAPTER 4

DEVELOPED PROTOTYPE

4.1 Detail Block Diagram

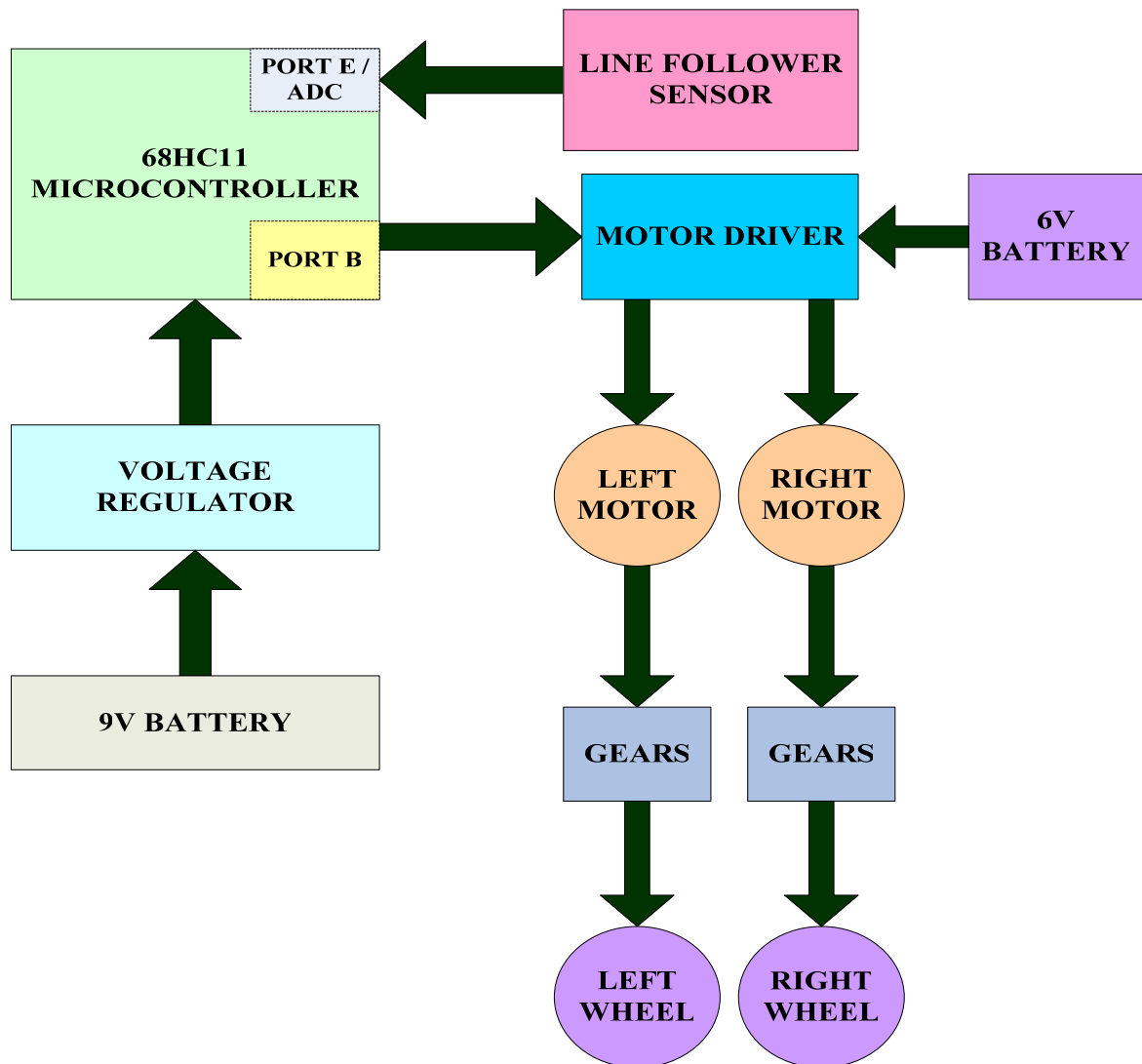


Figure 4.1 Block diagram for the whole system

The detail block diagram shows all the important parts of this prototype. The microcontroller unit controls all the circuitry and operation. The microcontroller is the MC68HC11A1 48 pin DIP. There are two power sources, the 9V and the 6V batteries. Since the microcontroller operating voltage is 5V, the 9V from the batteries is powered down to 5V by the voltage regulator 7805 that also maintain the fixed 5V to the microcontroller.

The line follower sensor acts as input to the microcontroller, so that the microcontroller will respond to the physical variables that it receives and make appropriate feedback according to the program inside. The left and right motors are the dc motors that function as the output for the microcontroller and there are motor drivers that prevent the 6V and high current from causing damage to the microcontroller. The dc motors operates in 6V batteries. Both of the dc motors and the driving wheels of the prototype provide the mechanical movement for the prototype.

The batteries are used instead of external power supply to let the prototype to move smoothly without any difficulties. It is more suitable to use batteries instead of external power supply although the motors use more voltage power to move the developed prototype.

4.2 Circuit Diagram

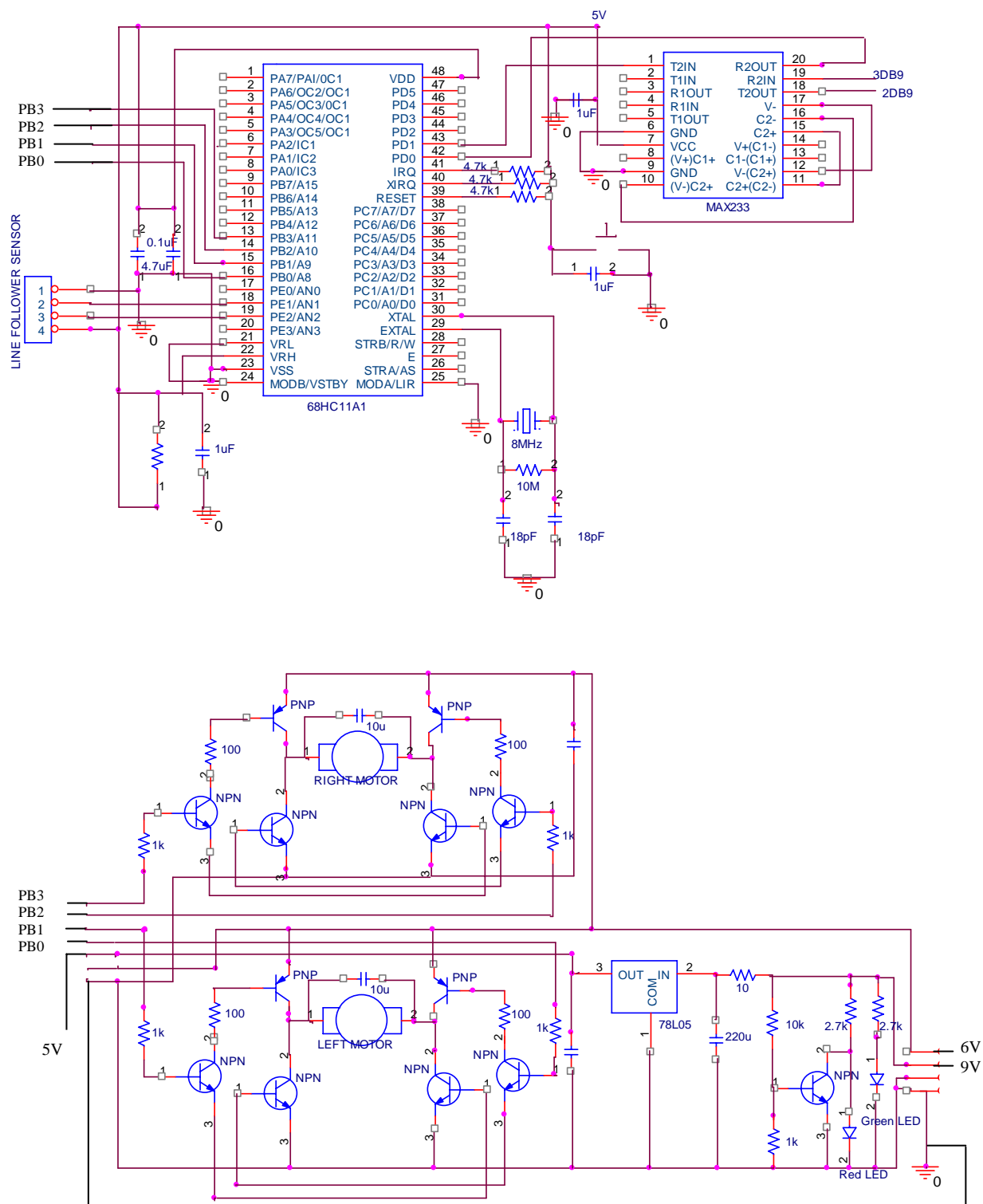


Figure 4.2 Circuit diagram for the whole system

The circuit diagram shows all the connection needed to build this prototype. It shows the microcontroller MC68HC11A1 in its bootstrap mode connection which is a simple one, the line follower sensor connection, the motor drivers and also the power from 9V and 6V batteries.

The line follower sensor is connected to GND, 5V supply, port E1 (PE1) and port E2 (PE2), as the port E of the microcontroller are functioning as analog to digital converter while the port B of the microcontroller are connected to the motors. Port B of the microcontroller can only be use as output port. There are 4 port B being used which is port B0, port B1, port B2, and port B3.

There are 2 H-Bridge circuits for each of the DC motors of each wheel. The operations for both of the H-Bridge circuit are the same. The green LED in the power circuitry is light up whenever the 9V power is connected or supplied to the whole circuit, and the red LED is light up whenever the 9V supply is low.

4.3 Algorithm

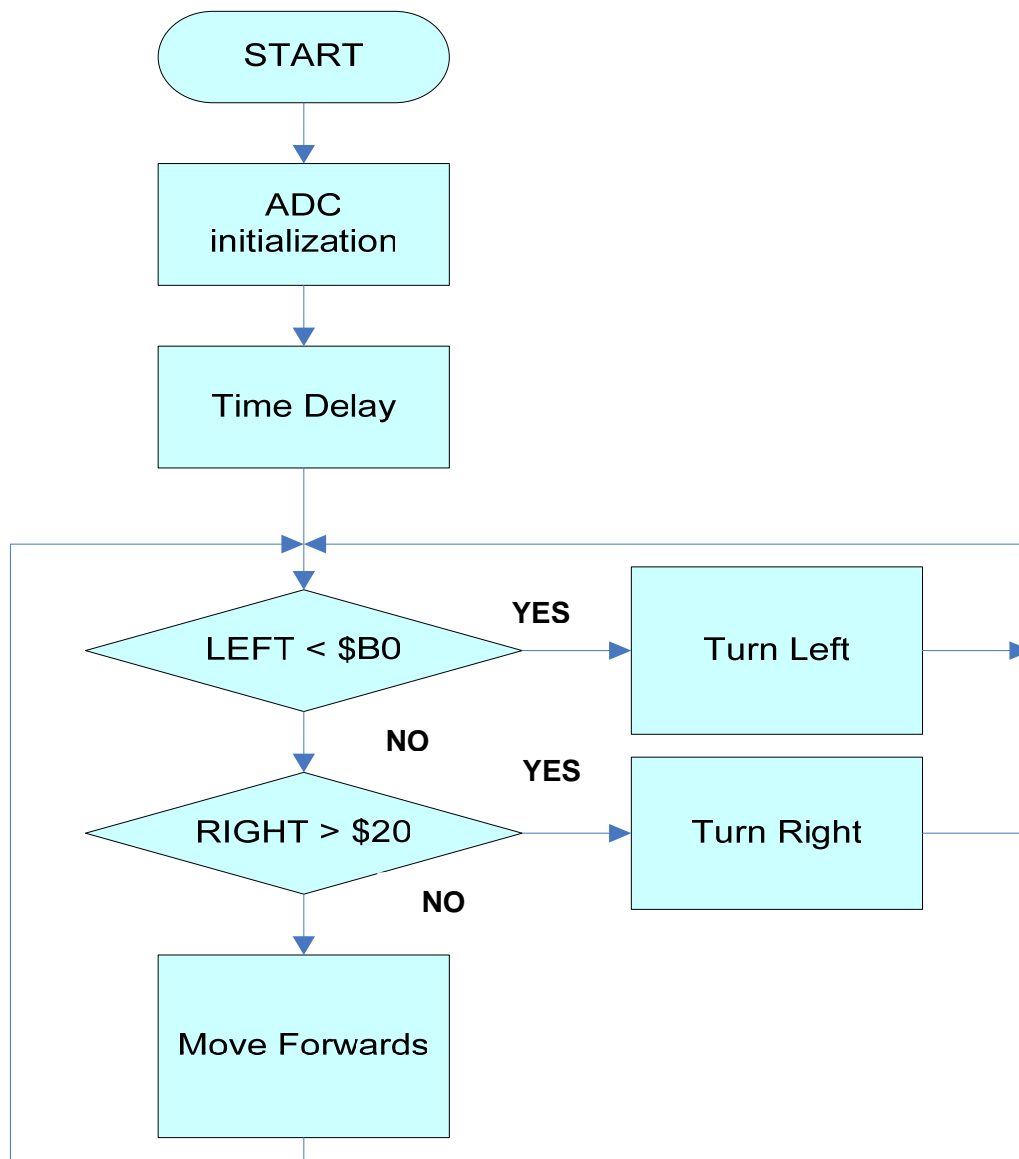


Figure 4.3 Flow chart for black line tracking

The flow chart shows the algorithm to let the prototype to move along a black line on a white background. The algorithm starts by the initialization of the port E of the microcontroller to be function as Analog-to-Digital Converter. The time delay of at least 100us is inserted after the ADCTL status register is initialized and before the result registers are read. Thus the margins value for both Left and Right units of the sensor are

being compared to the value in the result registers. The prototype will make the navigation according to the assembly language programming developed from the flow chart.

4.4 Final Prototype

In this section, some of the photos of the final prototype are shown to give close view. This final prototype is namely Nice-bot as it look nice and simple.

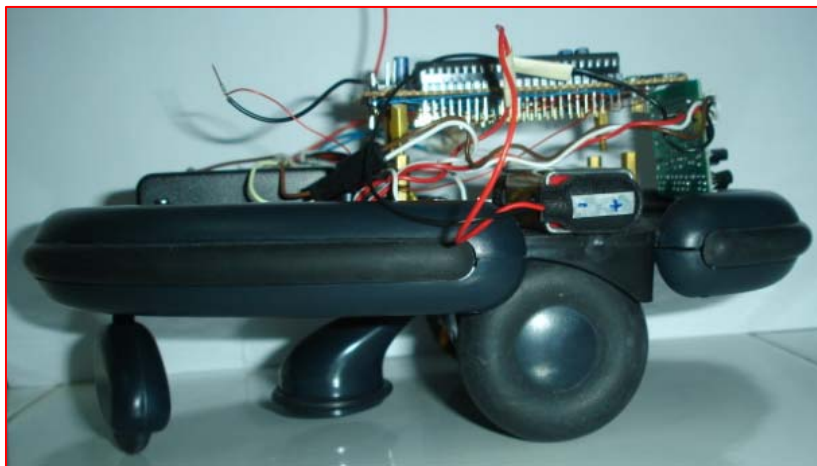


Figure 4.4 Left side view



Figure 4.5 right side view

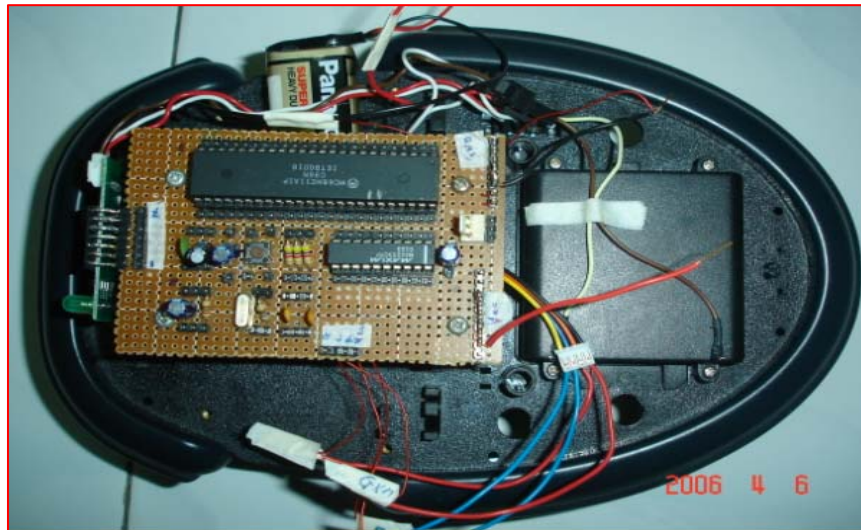


Figure 4.6 Top view



Figure 4.7 Bottom view



Figure 4.8 Front view



Figure 4.9 Rear view

4.5 List and cost of The components

Table 4.1 List and cost of components.

Components	Specifications	Quantity	Cost
Microcontroller	MC68HC11A1	1	RM30.00
RS232 communication	MAX 233	1	RM5.00
Resistors	4.7Kohm	3	RM0.50
	1Kohm	1	
	1Mohm	1	
Capacitors	4.7uF	1	RM2.10
	0.1uF	1	
	1uF	3	
	18pF	2	
Switch	Touch switch	1	RM0.60
DB9	Female	1	RM3.00
Oscillator	8MHz	1	RM0.50
Batteries	9V	1	RM9.40
	1.5V	4	RM5.40
Real Robot		5	RM 109.50
TOTAL			RM 166

The total estimation cost for this project to build the prototype is approximately RM166. It is a cost effective project that does not involved buying the expensive components. The amount is affordable.

CHAPTER 5

RESULTS AND DISCUSSIONS

This chapter discuss on the results obtained by the final prototype towards completing the objective of this project. The discussion can be divided into early result and implementation on realistic application.

5.1 Early Result

The early result is the first step that determines whether the objective of this project is successful or not. This includes the testing of the final prototype to move along path of black line tracking on white background.

5.1.1 Black Line Tracking

There are some characteristics of the line to consider with before a final tracking line is set up. Therefore, the final prototype is tested to move along a few shapes of curve of the black line to see its feedback. The black line tracking is made of black masking tape on white paper. These tests are important to know the capability of this prototype to move along the route, as well as these characteristics that obtained will be the guidelines for future line tracking work to be made for this prototype.

The voltage values from result registers ADR2 and ADR3 can be trace from the JBug software. These values from the result registers are recorded when the line follower sensor is aimed on black line and white paper surface. For this black line tracking test, the LEFT unit of the sensor is aimed on black surface while the RIGHT unit of the sensor is aimed on white surface. From the values shows by the result registers, the margin for LEFT unit sensor and RIGHT unit sensor are decided.

5.1.1.1 Smooth Curve, line width = 1.8cm



Figure 5.1 Smooth curve of black line with line width = 1.8 cm

The result that obtained is that, Nice-bot has no problem negotiating a smooth curve in both directions of clockwise and anticlockwise. It still can move along the route. But during the navigation, the Nice-bot can't make it for sure that it can maintain following the black line due to the line width is not so good enough.

5.1.1.2 Rounded 90 degrees bend, line width = 1.8cm



Figure 5.2 Rounded 90 degrees bend of black line with line width = 1.8 cm

The result that obtained is that the Nice-bot has no problem negotiating the rounded 90 degrees bend of black line either in clockwise or anticlockwise direction. It still can move along the route but during the navigation, the Nice-bot can't make it for sure that it can maintain following the black line due to the line width is not so good enough.

5.1.1.3 Sharp 90 degrees bend, line width = 1.8cm



Figure 5.3 Sharp 90 degrees bend of black line with line width = 1.8 cm

The result that obtained is that the Nice-bot has quite a problem to move sharp 90 degrees bend of black line with line width = 1.8 cm, as the width give not so much value for the line follower sensor to stay on the line. The Nice-bot can't move to make the 90 degrees of sharp turn of the black line with the line width is equal or below 1.8cm.

5.1.1.4 Sharp 90 degrees bend, line width = 2.6cm



Figure 5.4 Sharp 90 degrees bend of black line with line width = 2.6 cm

The result that obtained is that the Nice-bot has no problem to move sharp 90 degrees bend of black line with width = 2.6 cm, as the width is the sufficient width for the line follower sensor to stay on the line. The sensor still can take the voltage readings as the Nice-bot make a hard turn to stay on the line. For a frequent turn, this Nice-bot can possibly fail to make a perfect turn as it can't differentiate the readings of right and left unit sensor.

5.1.1.5 Sharp 90 degrees bend, line width = 3.6cm



Figure 5.5 Sharp 90 degrees bend of black line with line width = 3.6 cm

The result that obtained is that the Nice-bot has no problem to move sharp 90 degrees bend of black line with width = 3.6 cm, as the width is quite a large width for the line follower sensor to stay on the line. The sensor still can take the voltage readings as the Nice-bot can make an easy turn to stay on the line. Thus the possibility to fail turn is very low.

5.1.1.6 Result Summary

Table 5.1 Table of early result summary

Curve of black Line	Result
Smooth curve, line width = 1.8cm	Can't maintain to stay on the line
Rounded 90 degrees bend, line width = 1.8 cm	Can't maintain to stay on the line
Sharp 90 degrees bend, line width = 1.8 cm	Can't make the 90 degrees turn
Sharp 90 degrees bend, line width = 2.6 cm	Make a hard 90 degrees turn to stay on the line
Sharp 90 degrees bend, line width = 3.6 cm	Make an easy 90 degrees turn to stay on the line

5.2 Implementing Realistic Application

At this stage, it is valuable if this project can be implemented to the real world. Although this Nice-bot is quite a simple line following robot, it can be adjusted and developed to be use in any job that applicable in the future.

At first, a testing is made on the laboratory floor which is a quite dark cement floor. In this part, this Nice-bot is tested to whether it can move along a white line tracking on the dark laboratory floor. The white line tracking is made by the white masking tape. The white lines are varied in curve and are shown as Figure 5.6 and Figure 5.7 below.

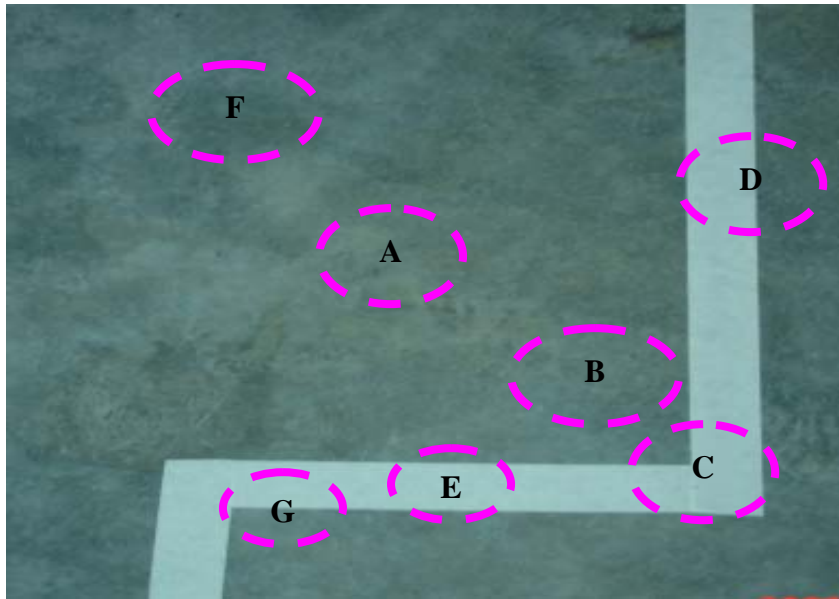


Figure 5.6 Sharp 90 degrees bend of white line on cement floor with line width = 3.6cm



Figure 5.7 Rounded 90 degrees bend of white line on dark cement floor with line width = 3.6cm

The values of result registers ADR2 and ADR3 are recorded as the Nice-bot with its line follower sensor is aimed on the white line as well as the dark cement floor.

Table 5.2 Hexadecimal values from result registers based on Figure 5.6

Circle of Area	RIGHT (PE2)	LEFT (PE1)
A	\$62	\$65
B	\$8D	\$8C
C	\$18	\$15
D	\$99	\$13
E	\$13	\$15
F	\$7E	\$7C
G	\$80	\$18

The hexadecimal values from result registers shown in table 5.2 are varies within its range. The values are traced by JBug software as the prototype is moving or turning on the laboratory floor. The prototype is being tested to whether it can follow the line on laboratory floor by taking the values of result registers whenever the line follower sensor is aimed on the surface. The left unit sensor is aimed on white line and connected to Port E1 (PE1) of the microcontroller while the right unit sensor is aimed on dark cement floor and connected to Port E2 (PE2) of the microcontroller.

From the values, the margins voltage values for RIGHT and LEFT units of the line follower sensor are determined. After that, the algorithm or flowchart and assembly language programming are developed for the Nice-bot to move along the white line route.

The result that obtained from the testing is that, the Nice-bot is able to move along the white line on the dark laboratory floor. It has no problem to follow the white line as the sensor can differentiate between the brightness of the line and the darkness of the cement floor.

Therefore, a track of white line can be made on the laboratory cement floor. Besides that, just like the results from the early result, the line width that best result in the Nice-bot to maintain following the line is 3.6 cm.

This prototype is exactly a success and reliable one and could move along white line on a not so black surface. Therefore, in the near future, the Nice-bot can be propose to be use as a simple line following robot to be use anywhere suitable such as in the laboratory, to clean up the floor, to carry things, and also a complex task, but a few adjustments to this Nice-bot model has to be made to make it more suitable for the jobs. The important characteristic to take note is that, this Nice-bot with its line following sensor are able to move along any dark and bright surfaces, since it can differentiate between the two surfaces.

CHAPTER 6

CONCLUSION AND SUGGESTIONS

6.1 Conclusion

As for conclusion, this project's objective to build an Automated Guided Vehicle model is successfully achieved. This prototype is able to move on a flat surface and move along black line on a white surface, depending on how much of the darkness and brightness of the surface as well as the width of the line. The microcontroller MC68HC11A1 can be use to control all the navigation of this prototype. The microcontroller is the important part that makes this prototype has quite a simple circuitry, as it already has complete internal circuitry in the chip. The line follower sensor and the DC motors are very reliable and suitable for the prototype. As the sensors gives continuous input to the microcontroller about the surface physical variables. The cost of this project is also effective and there are much profitable aspects if this prototype is developed in the future.

6.2 Suggestion for Future Work

There are many ideas and suggestions can be added towards developing this prototype in the future as this prototype could be as one of all the AGVs around the countries that help in human living.

As one of the suggestions is that, this prototype can be attached with more sensors to make it could work in much more complex task. There are many sensors nowadays, and the choices of sensors could be varied, but the selection of sensors must be applicable to this prototype. This model's chassis could be adjusted to make it suitable for future task,

and some other components could also be added. The adjustment may consider the factors of desired function to be built to the prototype, making it look more smart, interesting, and commercial. Other than that, the microcontroller DIP chip can be change to other type such as PLCC that could save the space of the board and also seem to look smarter.

Therefore, this prototype has the potential and could be the basic to the future adjustment of this prototype to work in the workplaces or in helping the residents to do their housework.

REFERENCES

1. Motorola Inc (1991). The MC68HC11A8 Technical Data. MC68HC11A8/D Rev. 6
2. Motorola Inc (1999). M68HC11 Bootstrap Mode. AN1060/D Rev. 1.0
3. Motorola Inc (2000). M68HC11 Eeprom Programming. AN1010/D
4. Miller, Gene H. (1999). "Microcomputer Engineering." New Jersey, Prentice Hall.
5. Spasov, Peter. (2004). "Microcontroller Technology." New Jersey, Prentice Hall.
6. Fredrick M. Cady. (1997). "Software and Hardware Engineering Motorola M68HC11." Oxford University Press.
7. ranchbots
 URL: http://www.ranchbots.com/line_follower/line_follower.htm
 Accessed on: March 30, 2006
8. Robot Room – sandwich, The Line-Following Robot
 URL: <http://www.robotroom.com/Sandwich.html>
 Accessed on: March 30, 2006
9. N. Tschichold-Gurman, S. J. Vestli, G. Schweitzer. "Operating Experiences with the Service Robot MOPS." Institute of Robotics, ETH Swiss Federal Institute of Technology, EUROBOT'99, ETH Zurich, Switzerland, Sept. 6-8, 1999.

10. Lpilsley

URL: <http://lpilsley.co.uk/cybot/>

Accessed on: January 5, 2006

11. Real Robots

URL: <http://www.realrobots.co.uk/>

Accessed on: October 1, 2005

12. Frog Navigation Systems: automated guided vehicles systems (AGVs) / Fahrerlose Transport System

URL: <http://www.frog.nl/>

Accessed on: March 20, 2006

13. Automated Guided Vehicle – Wikipedia, the free encyclopedia

URL: http://en.wikipedia.org/wiki/Automated_Guided_Vehicle

Accessed on: January 12, 2006

14. Line Follower Robot

URL: <http://www.kmitl.ac.th/~kswichit/LFrobot/LFrobot.htm>

Accessed on: August 15, 2005

15. Safety Practices for Automated Guided Vehicles (AGVs)

URL: http://www.asse.org/prac_spec_tech2.htm

Accessed on: August 15, 2005

16. Pilot project

URL: <http://faculty.washington.edu/jbs/itrans/parkshuttle.htm>

Accessed on: March 30, 2006

APPENDICES

APPENDIX A: TECHNICAL DATA OF MC68HC11

Technical Summary

8-Bit Microcontrollers

1 Introduction

The MC68HC11A8, MC68HC11A1, and MC68HC11A0 high-performance microcontroller units (MCUs) are based on the M68HC11 Family. These high speed, low power consumption chips have multiplexed buses and a fully static design. The chips can operate at frequencies from 3 MHz to dc. The three MCUs are created from the same masks; the only differences are the value stored in the CONFIG register, and whether or not the ROM or EEPROM is tested and guaranteed.

For detailed information about specific characteristics of these MCUs, refer to the *M68HC11 Reference Manual* (M68HC11RM/AD).

1.1 Features

- M68HC11 CPU
- Power Saving STOP and WAIT Modes
- 8 Kbytes ROM
- 512 Bytes of On-Chip EEPROM
- 256 Bytes of On-Chip RAM (All Saved During Standby)
- 16-Bit Timer System
 - 3 Input Capture Channels
 - 5 Output Compare Channels
- 8-Bit Pulse Accumulator
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog System
- Synchronous Serial Peripheral Interface (SPI)
- Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
- 8-Channel, 8-Bit Analog-to-Digital (A/D) Converter
- 38 General-Purpose Input/Output (I/O) Pins
 - 15 Bidirectional I/O Pins
 - 11 Input-Only Pins and 12 Output-Only Pins (Eight Output-Only Pins in 48-Pin Package)
- Available in 48-Pin Dual In-Line Package (DIP) or 52-Pin Plastic Leaded Chip Carrier (PLCC)

Table 1 MC68HC11Ax Family Members

Device Number	ROM	EEPROM	RAM	CONFIG*	Comments
MC68HC11A8	8K	512	256	\$0F	Family built around this device
MC68HC11A1	0	512	256	\$0D	ROM disabled
MC68HC11A0	0	0	256	\$0C	ROM and EEPROM disabled

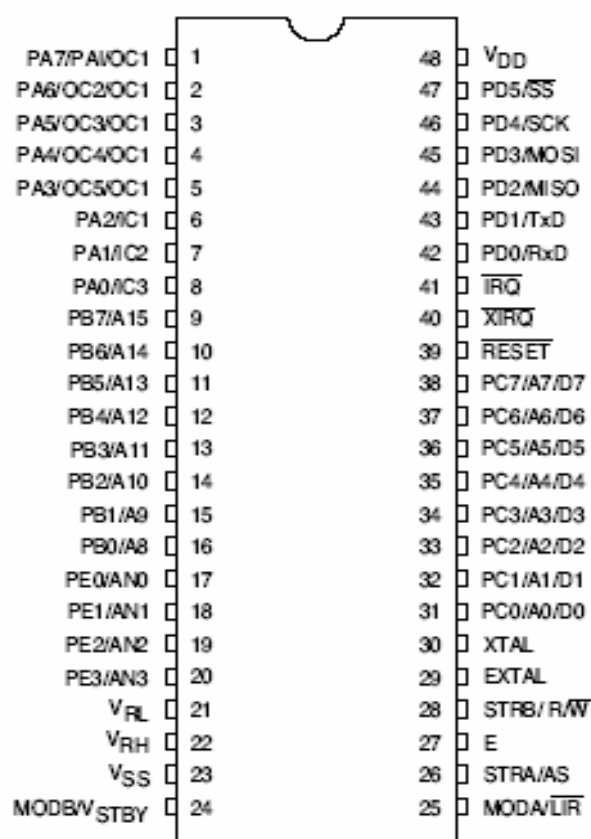


Figure 3 48-Pin DIP Pin Assignments

2.1 Memory Maps

Memory locations are the same for expanded multiplexed and single-chip modes. The on-board 256-byte RAM is initially located at \$0000 after reset. The 64-byte register block originates at \$1000 after reset. RAM and/or the register block can be placed at any other 4K boundary (\$x000) after reset by writing an appropriate value to the INIT register. The 512-byte EEPROM is located at \$B600 through \$B7FF after reset if it is enabled. The 8 Kbyte ROM is located at \$E000 through \$FFFF if it is enabled.

Hardware priority is built into the memory remapping. Registers have priority over RAM, and RAM has priority over ROM. The higher priority resource covers the lower, making the underlying locations inaccessible.

In special bootstrap mode, a bootloader ROM is enabled at locations \$BF40 through \$BFFF.

In special test and special bootstrap modes, reset and interrupt vectors are located at \$BFC0 through \$BFFF.

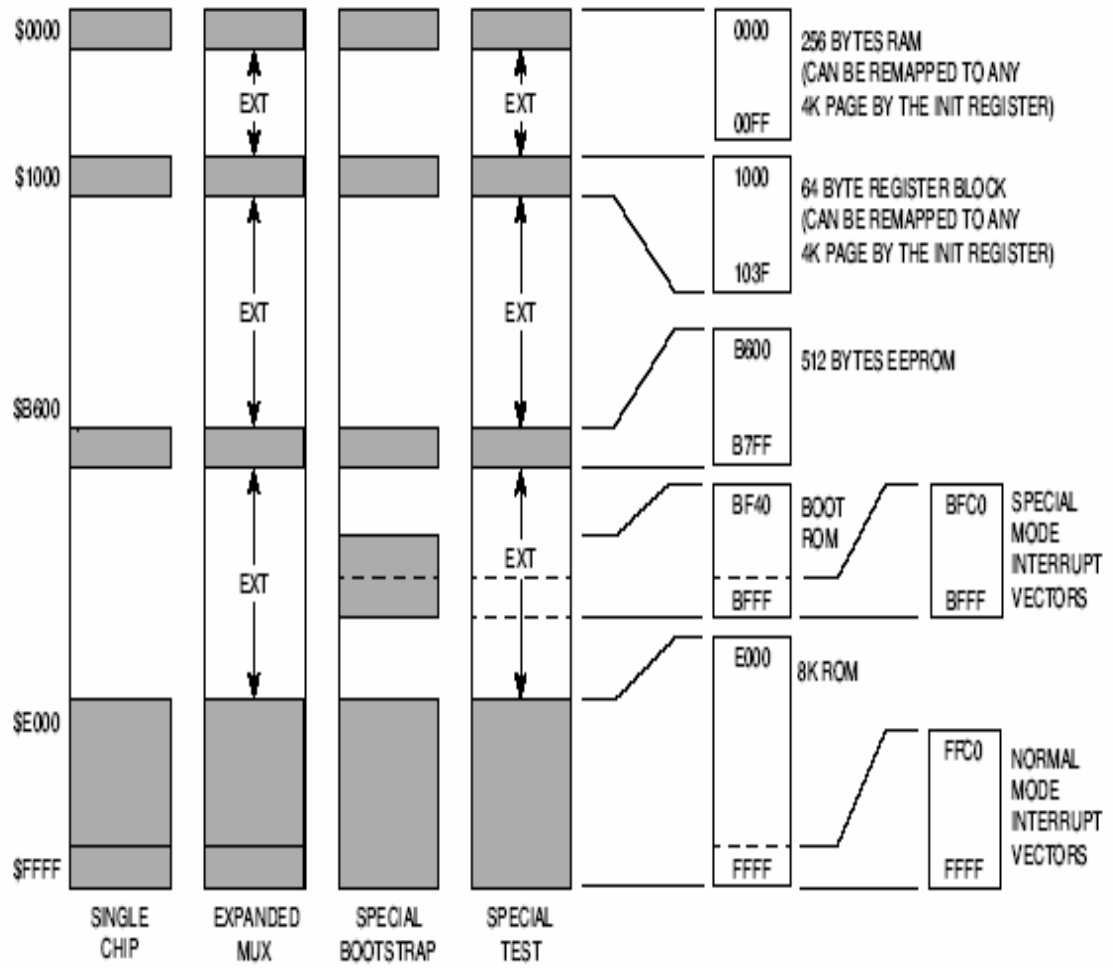


Figure 5 Memory Map

4 Electrically Erasable Programmable Read-Only Memory (EEPROM)

The 512 bytes of EEPROM in the MC68HC11A8 are located at \$B600 through \$B7FF. The EEON bit in CONFIG controls the presence or absence of the EEPROM in the memory map. When EEON = 1 (erased state), the EEPROM is enabled. When EEON = 0, the EEPROM is disabled and out of the memory map. EEON is reset to the value last programmed into CONFIG. An on-chip charge pump develops the high voltage required for programming and erasing. When the E clock is less than 1 MHz, select an internal clock. This drives the EEPROM charge pump by writing a one to the CSEL bit in the OPTION register.

The PPROG register controls the programming and erasing of the EEPROM. To erase the EEPROM, complete the following steps using the PPROG register:

1. Write to PPROG with the ERASE, EELAT, and appropriate BYTE and ROW bits set.
2. Write to the appropriate EEPROM address with any data. Row erase only requires a write to any location in the row. Bulk erase is accomplished by writing to any location in the array.
3. Write to PPROG with ERASE, EELAT, EEPGM, and the appropriate BYTE and ROW bits set.
4. Delay for 10 ms or more, as appropriate.
5. Clear the EEPGM bit in PPROG to turn off the high voltage.
6. Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

To program the EEPROM, complete the following steps using the PPROG register:

1. Write to PPROG with the EELAT bit set.
2. Write data to the desired address.
3. Write to PPROG with the EELAT and EEPGM bits set.
4. Delay for 10 ms or more, as appropriate.
5. Clear the EEPGM bit in PPROG to turn off the high voltage.
6. Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

PPROG — EEPROM Programming Control

\$103B

	Bit 7	6	5	4	3	2	1	Bit 0
	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM
RESET:	0	0	0	0	0	0	0	0

ODD — Program Odd Rows in Half of EEPROM (TEST)

EVEN — Program Even Rows in Half of EEPROM (TEST)

BYTE — Byte/Other EEPROM Erase Mode

The BYTE bit overrides the ROW bit.

0 = Row or bulk erase mode is used

1 = Erase only one byte of EEPROM

ROW — Row/All EEPROM Erase Mode

The ROW bit is only valid when BYTE = 0.

0 = All 512 bytes of EEPROM are erased

1 = Erase only one 16-byte row of EEPROM

BYTE	ROW	Action
0	0	Bulk Erase (All 512 Bytes)
0	1	Row Erase (16 Bytes)
1	0	Byte Erase
1	1	Byte Erase

ERASE — Erase/Normal Control for EEPROM

0 = Normal read or program mode

1 = Erase mode

EELAT — EEPROM Latch Control

0 = EEPROM address and data bus configured for normal reads

1 = EEPROM address and data bus configured for programming or erasing

EEPGM — EEPROM Program Command

0 = Programming or erase voltage switched off to EEPROM array

1 = Programming or erase voltage switched on to EEPROM array

CONFIG — COP, ROM, EEPROM Enables

\$103F

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	NOSEC	NOCOP	ROMON	EEON
RESET:	0	0	0	0	—	—	—	—

NOTE

The bits of this register are implemented with EEPROM cells. Programming and erasing follow normal EEPROM procedures. The erased state of CONFIG is \$0F. A new value is not readable until after a subsequent reset sequence. CONFIG can only be programmed or erased in special modes.

NOSEC — EEPROM Security Disable

NOSEC has no meaning unless the security mask option was specified before the MCU was manufactured.

0 = Security enabled (available as a mask option on MC68HC11A8 only)

1 = Security disabled

NOCOP — COP system disable

Refer to 3 Resets and Interrupts.

ROMON — ROM Enable

Refer to 2 Operating Modes and Memory Maps.

EEON — EEPROM Enable

0 = EEPROM is removed from the memory map

1 = EEPROM is present in the memory map

5 Parallel Input/Output

The MC68HC11A8 has up to 38 input/output lines, depending on the operating mode. Port A has three input-only pins, four output-only pins, and one bidirectional I/O pin. Port A shares functions with the timer system.

Port B is an 8-bit output-only port in single-chip modes and is the high-order address in expanded modes.

Port C is an 8-bit bidirectional port in single-chip modes and the multiplexed address and data bus in expanded modes.

Port D is a 6-bit bidirectional port that shares functions with the serial systems.

Port E is an 8-bit input-only port that shares functions with the A/D system.

Simple and full handshake input and output functions are available on ports B and C lines in single-chip mode. A description of the handshake functions follows.

In port B simple strobed output mode, the STRB output is pulsed for two E-clock periods each time there is a write to the PORTB register. The INVB bit in the PIOC register controls the polarity of STRB pulses.

In port C simple strobed input mode, port C levels are latched into the alternate port C latch (PORTCL) register on each assertion of the STRA input. STRA edge select, flag and interrupt enable bits are located in the PIOC register. Any or all of the port C lines can still be used as general purpose I/O while in strobed input mode.

Port C full handshake mode involves port C pins and the STRA and STRB lines. Input and output handshake modes are supported, and output handshake mode has a three-stated variation. STRA is an edge detecting input, and STRB is a handshake output. Control and enable bits are located in the PIOC register.

In full input handshake mode, the MCU uses STRB as a "ready" line to an external system. Port C logic levels are latched into PORTCL when the STRA line is asserted by the external system. The MCU then negates STRB. The MCU reasserts STRB after the PORTCL register is read. A mix of latched inputs, static inputs, and static outputs is allowed on port C, differentiated by the data direction bits and use of the PORTC and PORTCL registers.

In full output handshake mode, the MCU writes data to PORTCL, which in turn asserts the STRB output to indicate that data is ready. The external system reads port C (the STRB output) and asserts the STRA input to acknowledge that data has been received.

In the three-state variation of output handshake mode, lines intended as three-state handshake outputs are configured as inputs by clearing the corresponding DDRC bits. The MCU writes data to PORTCL and asserts STRB. The external system responds by activating the STRA input, which forces the MCU to drive the data in PORTCL out on all of the port C lines. This mode variation does not allow part of port C to be used for static inputs while other port C pins are being used for handshake outputs. Refer to the PIOC register description.

PORTB — Port B Data

\$1004

	Bit 7	6	5	4	3	2	1	Bit 0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
S. Chip or Boot:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET:	0	0	0	0	0	0	0	0
Expan. or Test:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8

PORTE — Port E Data

\$100A

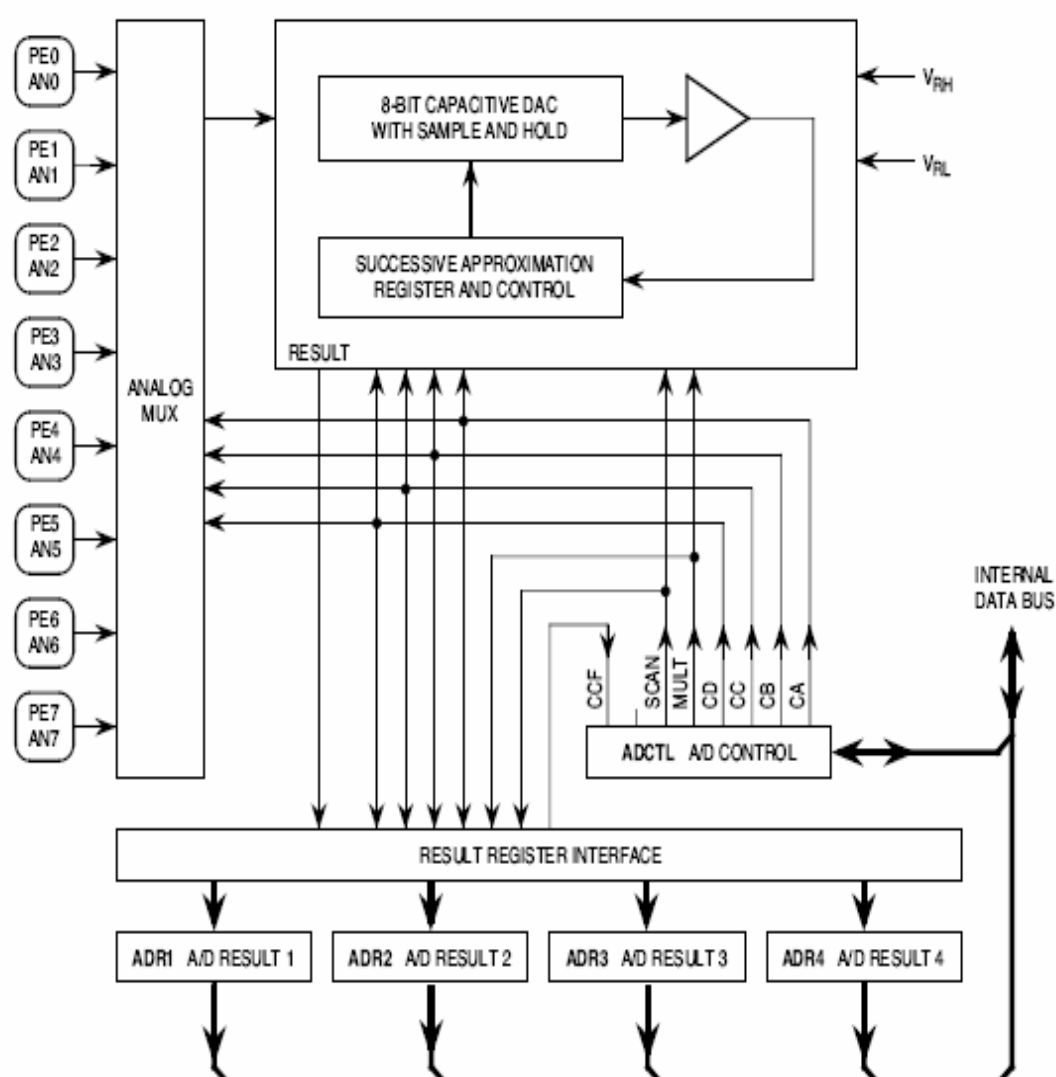
	Bit 7	6	5	4	3	2	1	Bit 0
	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:	U	U	U	U	U	U	U	U
Alt. Pin Func.:	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

10 Analog-to-Digital Converter

The A/D converter system uses an all capacitive charge redistribution technique to convert analog signals to digital values. The MC68HC11A8 A/D system is an 8-channel, 8-bit, multiplexed-input, successive-approximation converter and is accurate to ± 1 least significant bit (LSB). It does not require external sample and hold circuits because of the type of charge redistribution technique used.

Dedicated lines V_{RH} and V_{RL} provide the reference supply voltage inputs. Refer to the A/D converter block diagram.

A multiplexer allows the single A/D converter to select one of 16 analog signals, as shown in the ADCTL register description.



ES0 AD BLOCK

Figure 13 A/D Converter Block Diagram

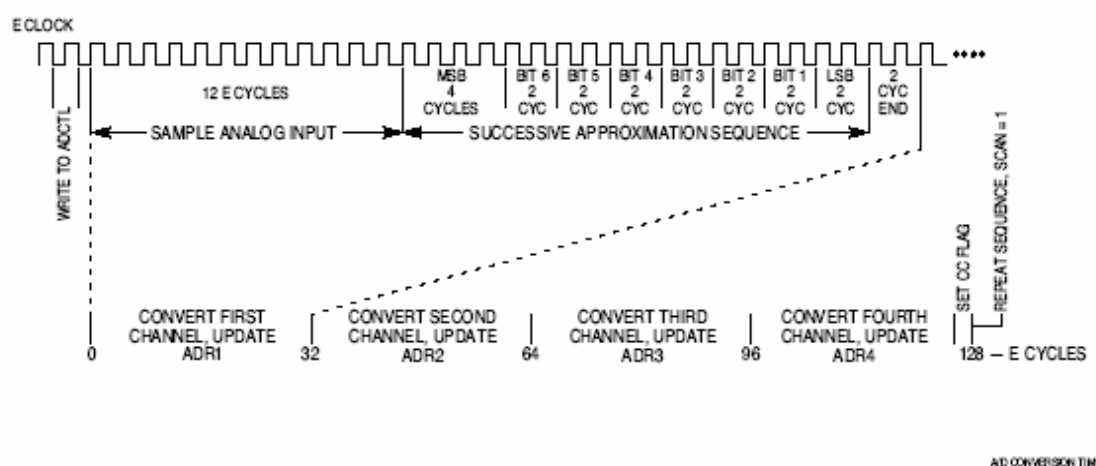


Figure 14 A/D Conversion Sequence

ADCTL — A/D Control/Status

\$1030

	Bit 7	6	5	4	3	2	1	Bit 0
	CCF	0	SCAN	MULT	CD	CC	CB	CA
RESET:	U	0	U	U	U	U	U	U

CCF — Conversions Complete Flag

Set after an A/D conversion cycle. Cleared when ADCTL is written.

SCAN — Continuous Scan Control

0 = Do four conversions and stop

1 = Convert four channels in selected group continuously

MULT — Multiple Channel/Single Channel Control

0 = Convert single channel selected

1 = Convert four channels in selected group

CD—CA — Channel Select D through A

Table 10 A/D Converter Channel Assignments

Channel Select Control Bits				Channel Signal	Result in ADRx if MULT = 1
CD	CC	CB	CA		
0	0	0	0	AN0	ADR1
0	0	0	1	AN1	ADR2
0	0	1	0	AN2	ADR3
0	0	1	1	AN3	ADR4
0	1	0	0	AN4*	ADR1
0	1	0	1	AN5*	ADR2
0	1	1	0	AN6*	ADR3
0	1	1	1	AN7*	ADR4
1	0	X	X	Reserved	ADR1–ADR4
1	1	0	0	V_{RH}^{**}	ADR1
1	1	0	1	V_{RL}^{**}	ADR2
1	1	1	0	$(V_{RH})/2^{**}$	ADR3
1	1	1	1	Reserved**	ADR4

* Not available in 48-pin package

**Used for factory testing

ADR1–ADR4 — A/D Results

\$1031–\$1034

	Bit 7	6	5	4	3	2	1	Bit 0	
\$1031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
\$1032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
\$1033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
\$1034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4

Table 11 Analog Input to 8-Bit Result Translation Table

	Bit 7	6	5	4	3	2	1	Bit 0
% (1)	50%	25%	12.5%	6.25%	3.12%	1.56%	0.78%	0.39%
Volts (2)	2.500	1.250	0.625	0.3125	0.1562	0.0781	0.0391	0.0195

(1) % of $V_{RH}-V_{RL}$ (2) $V_{RL} = 0.0\text{ V}$; $V_{RH} = 5.0\text{ V}$

OPTION — System Configuration Options

\$1039

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	0	CR1*	CR0*
RESET:	0	0	0	1	0	0	0	0

*Can be written only once in first 64 cycles out of reset in normal modes, or any time in special modes.

ADPU — A/D Power Up

0 = A/D Converter powered down

1 = A/D Converter powered up

CSEL — Clock Select

0 = A/D and EEPROM use system E clock

1 = A/D and EEPROM use internal RC clock

APPENDIX B: LISTING FILE AND SOURCE CODE

File LINEPR~1.ASM p By ASM11 v1.84d [Tuesday, March 20, 2006 4:11 pm]

```

1      B600                                ORG      $B600
2
3 B600: 8680                                [ 2]      LDAA      #$80
4 B602: B710 39                            [ 4]      STAA      $1039
5 B605: 8630                                [ 2]      LDAA      #$30
6 B607: B710 30                            [ 4]      STAA      $1030
7 B60A: 8D29 (B635)                        [ 6]      BSR       DELAY
8
9 B60C: 8660                                [ 2]      AGAIN    LDAA      #$60
10 B60E: B110 32                           [ 4]      CMPA      $1032
11 B611: 2310 (B623)                       [ 3]      BLS       LEFT
12
13 B613: 8630                                [ 2]      LDAA      #$30
14 B615: B110 33                           [ 4]      CMPA      $1033
15 B618: 2412 (B62C)                       [ 3]      BHS       RIGHT
16
17 B61A: 860A                                [ 2]      STR      LDAA      #$0A
18 B61C: B710 04                           [ 4]      STAA      $1004
19 B61F: 8D14 (B635)                       [ 6]      BSR       DELAY
20 B621: 20E9 (B60C)                       [ 3]      BRA       AGAIN
21
22 B623: 8609                                [ 2]      LEFT     LDAA      #$09
23 B625: B710 04                           [ 4]      STAA      $1004
24 B628: 8D0B (B635)                       [ 6]      BSR       DELAY
25 B62A: 20E0 (B60C)                       [ 3]      BRA       AGAIN
26
27 B62C: 8606                                [ 2]      RIGHT    LDAA      #$06
28 B62E: B710 04                           [ 4]      STAA      $1004
29 B631: 8D02 (B635)                       [ 6]      BSR       DELAY
30 B633: 20D7 (B60C)                       [ 3]      BRA       AGAIN
31
32 B635: C6FF                                [ 2]      DELAY    LDAB      #$FF
33 B637: 5A                                [ 2]      LAGI     DECB
34 B638: 26FD (B637)                       [ 3]      BNE      LAGI
35 B63A: 39                                [ 5]      RTS

```

S E G M E N T U S A G E R E P O R T							
Segment	Start	End	Size	CodObj	DatObj	TotObj	Lines
-----	-----	-----	-----	-----	-----	-----	-----
ROM	\$B600	\$B63A	\$003B	\$003B	\$0000	\$003B	35
Summary	\$B600	\$B63A	\$003B	\$003B	\$0000	\$003B	35

O V E R A L L M E M O R Y U S A G E

Total RAM size:	0	\$0000	0.00 KB	
Total CODE size:	59	\$003B	0.06 KB	(27
instructions)				
Total DATA size:	0	\$0000	0.00 KB	
Total IMAGE size:	59	\$003B	0.06 KB	
Lowest address :	46592	\$B600		
Highest address :	46650	\$B63A		
Address Range :	59	\$003B	0.06 KB	(Used Range:
100.00%)				

Number of INCLUDE files: 0

Assembled 35 lines (No Errors, Warnings: 0)

*** End of LINEPR~1.ASM listing ***

APPENDIX C: ASM FILE

```

ORG    $B600

LDAA   #$80
STAA   $1039
LDAA   #$30
STAA   $1030
BSR    DELAY

AGAIN  LDAA   #$60
      CMPA   $1032
      BLS    LEFT

      LDAA   #$30
      CMPA   $1033
      BHS    RIGHT

STR    LDAA   #$0A
      STAA   $1004
      BSR    DELAY
      BRA    AGAIN

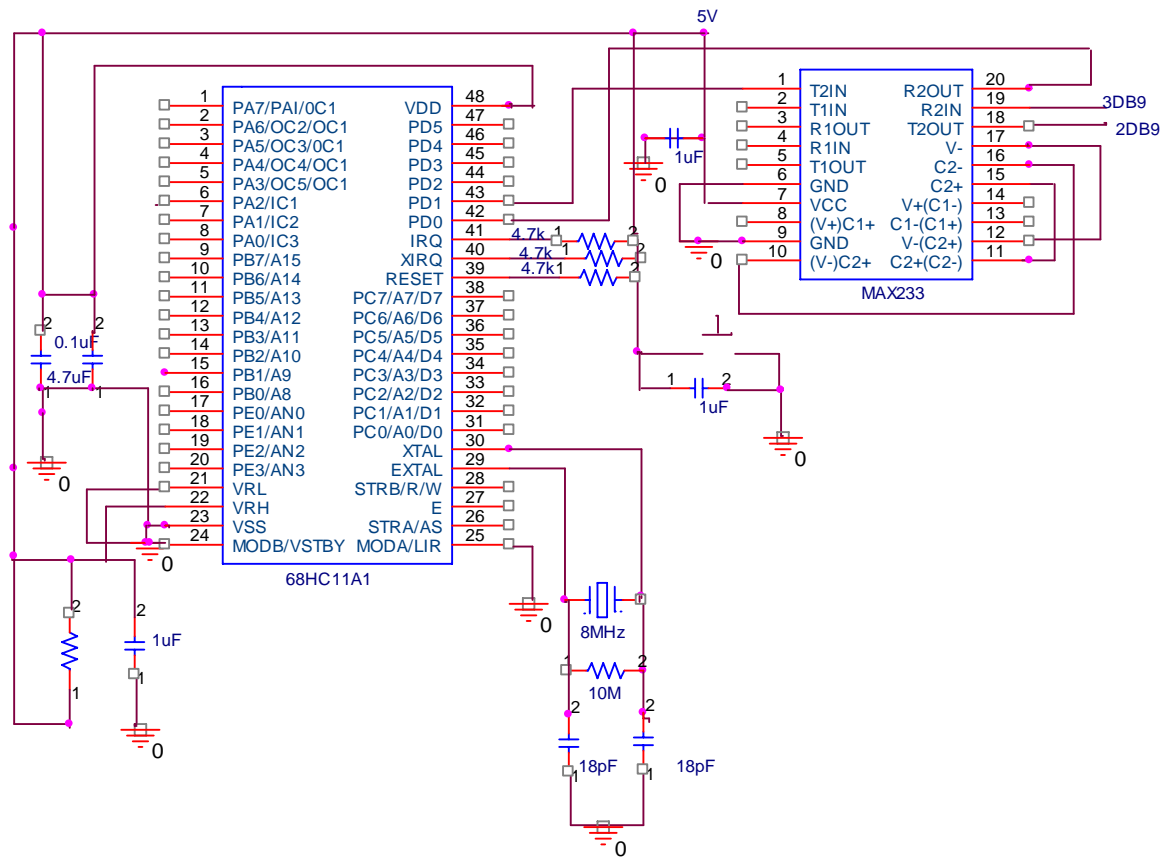
LEFT   LDAA   #$09
      STAA   $1004
      BSR    DELAY
      BRA    AGAIN

RIGHT  LDAA   #$06
      STAA   $1004
      BSR    DELAY
      BRA    AGAIN

DELAY  LDAB   #$FF
LAGI   DECB
      BNE    LAGI
      RTS

```

APPENDIX D: BASIC CONNECTION OF BOOTSTRAP MODE



APPENDIX E: CD CONTENT

- 1) FinalThesis.pdf**
- 2) ASM Files**
- 3) JBug Software**
- 4) Photos**
- 5) Video**