CERTIFICATION OF APPROVAL

ROBOTIC LAWN MOWER

by

TIMOTHY THIEN CHING KAE

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Dr Taj Mohammad Baloch
Senior Lecturer
Electrical & Electronic Engineering
Universiti Teknologi PETRONAS

Dr. Taj Mohammad Baloch
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

TIMOTHY THIEN CHING KAE

ABSTRACT

The Final Year Project course is designed for students to do research; design and development work in each discipline, to produce practical solutions. It provides opportunity for students to use the tools and techniques of problem-solving by engagement of the project. Under proper guidance of supervisor, the students will shape the direction in the field of interest as a preparation for approaching their desired career path in the near future, as well as gain better understanding of the responsibilities they have to shoulder when they undertake a project.

The objective of this project is to produce an automatic robotic lawn mover which is able to cut grass of a specified area of flat land. The first approach to this project is to conduct intensive literature reviews regarding the functions of robots and robotic designs and programming. The Second phase of this project is implementation of the theory to build a functioning robot.

The scope of the study would focus on the design and implementation of the robot from scratch. The study is broken down into sub sections, which are electronic circuits, movement mechanism and programming. These subsections are developed and combined until the implementation of the workable robot.

In the discussion section, all findings would be discussed in more detail and alternatives are compared as to assure the objectives are met during implementation.

ACKNOWLEDGEMENTS

First and foremost, the author would like to take this opportunity to express their appreciation to all the parties that is involved in making Final Year Design Project success. The undergoing of this project would not have been possible without the assistance and guidance of certain individuals and organization whose contributions have helped in its completion.

The author would like to express his deepest appreciation to his project supervisor, Dr. Taj Mohammad Baloch for his endless support, guidance and consultation regarding this project throughout the year.

The author also would like to thank Mr. Mohd Haris Md Khir and Dr. Naufal for all their guidance and the opportunity to be part of the ROBOCON 2006 team of University Technology PETRONAS (UTP). Greatest gratitude goes to all the ROBOCON 2006 team members of University Technology PETRONAS (UTP) for their reference and support within the project time scope

Special thanks to Ms Siti Hawa who has given countless technical support to the author as a lab technician. She has been helpful in giving advices to the author, never failing in providing the electrical component and parts needed for this project to the author.

Last but not least, the author would like to thanks Mr. Yeap Yeow Chong for providing the mechanical drawings using the software Catia ® and not forgetting to all those who have contributed directly or indirectly to the success off this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND

In the information age, robots are widely used in small and wide scale Industries. The robots are used to reduce human dependencies in performing a task and to reduce time for production while maintaining consistent quality of production outputs. Robot precision and accuracy allows task such as Processor Chips fabrication possible, in which transistors placed between each other in nano-meter length. Robotic devices are not just used in the manufacturing industries; it is also used for domestic living such as “smart houses” and advance surveillance systems.

1.2 PROBLEM STATEMENT

Malaysia is blessed with the tropical weather which allows plants to grow all year long. Many Malaysians own houses which includes a compound often used as garden. Grass which grows everyday is needed to be maintained in a short length. Failure to do so would cause declination in aesthetic aspects of the garden. However, lawn moving is often a hassle and time consuming.

The task of manual lawn moving could be replaced automatically with the help of a robot programmed to perform such task. However, Robots are often expensive and costly. This factor is often the case of discouraging people from being involved in the robotics field. The success of building an inexpensive automatic lawn mover would allow house owners to maintain their lawn without spending much time and money doing so.

The robot must be an autonomous robot where it can mow lawn of a given specific area of the garden without human supervision. These are some of the criteria set for the mobile robot:

1. The robot must not go out of the given scope of work area.
2. The robot must not leave any spot of the given area unattended
3. The robot must be able to avoid collision of obstacles along its pathway.
4. The robot must not be too costly and be affordable

1.3 OBJECTIVES

The specific objective of this project is to develop a robot which would be able to help users in domestic living. For this case which is an automatic lawn mover robot. The robot is able to cover a specific area of flat land and is able to avoid any obstacle along the way. The robot should be moderately cheap and easy to implement with the help of the PIC microcontroller or PLC controller. If the PIC microcontroller is used, the circuitry should be easy to understand to arise the interest in university students in further development and implementation of robots as university projects or simply a personal hobby.

An estimated period of 1 month would be needed for preliminary research. The remaining months would be used for developing the PIC program and building the robot.

There are a few objectives need to be achieved by the end of the project completed. The objective will be stated clearly as to make sure the success of the project implemented. The objectives of this project are:

1.3.1 To design a simple structure and movement mechanism for the robot

The design of the structure and movement mechanism of the robot is desired to be simply yet effective for easy troubleshooting purpose. The simple mechanism can also reduce the complexity of the programming of the robot and reduce cost.

1.3.2 To set up the boundaries of the working area of the robot

Setting up the boundaries of work area is essential for the robot in order to be able to meet its objective; which is to mow the lawn of a target area. Robot moving out of the specified boundary would cause a waste of time and resource.

1.3.3 To be able to avoid obstacle collision

The robot is desired to be able to take proactive measures of avoiding collision of obstacles along its path. Failure to do so would damage to the robot and the robot stagnant on that position due to the obstacle blocking its way thus, causing the objective of the robot unmet.

1.4 SCOPE OF STUDY

The scope of study can be divided into 3 subsections. In which interdependency and coordination between all three subsection must be developed to produce a functioning mobile robot

1.4.1 Controllers and Programming

Controllers are the brain of the robot. There are various choices of controllers to choose from. However, the cost is a heavy factor for this project. Therefore, PIC micro-controller has been chosen to enable the robot to be cost effective. The programming languages required for this controller are the Assembly language and C Programming

1.4.2 Electrical Circuits

Suitable sensors are needed for collision avoidance system. Other electrical circuitry is required to enable the proper function of the mechanical devices and to supply power for the robot.

1.4.3 Robotic Movement and Mechanism

Motors and actuator are the “muscles” of a robot. Proper motors need to be selected for desired output and tasks. Three types of motor have been identified for the robot; Continuous Motors, Stepper Motors and Servo Motors.

CHAPTER 2
LITERATURE REVIEW

2.1 OVERVIEW

Robots may include feedback driven connection between sense and action, not under direct human control. Responses may take form of operation of electro-magnetic motor or actuators (effectors) which may control arm, open or close grips. Control and feedback is provided by computer program which can be either external or internally inside the robot itself.

There are two basic ways of using effectors, in which the first, to move the robot around (locomotion) and the second, to move other object around (manipulation). The distinction divides the robotics into two mostly separate categories: mobile robots (moving) and manipulator robotics (grabbing)

2.2 HISTORY

The idea of artificial people dates back as the ancient legend of Cadmus, who sowed dragon teeth that turned into soldiers, and the myth of Pygmalion, whose statue of Galatea came to life. According to classical mythology, the deformed god of metalwork, Hephaestus, created mechanical servants, raging from intelligent, golden handmaidens to utilitarian three legged tables that could move about under their own power. [7]

Czech writer Karel Capek introduced the word "ROBOT" in his play RUR (Rossuum's Universal Robots) in 1921, "Robot: in Czech comes from the word "robota" which means "compulsory labour". The earliest ideas that could be related to robotic is in 350 B.C by a Greek mathematician, Archytas. He created a mechanical bird he called "The Pigeon: which was propelled by steam. [7]

The fist recorded design of a humanoid robot was done by Leonardo da Vinci (1495). Da Vinci 's notebook contained detailed drawing for a mechanical knight that was apparently able to sit up, wave its arms and move its head and jaw.

The first known functioning robot was created in 1738 by Jacques de Vaucanson, who made an android that played the flute. Many considered the first robot in the modern sense to be a teleoperated boat, similar to a modern ROV (Remote Operated Vehicle), designed by Nikola Tesla and demonstrated at an 1898 exhibition in Madision Square Garden. In the thirties, Westinghouse made a humanoid robot known as Electro, which was exhibited at the 1939 and 1940 World's Fairs. The first electronic autonomous robot was created by Grey Walter at Bristol University England in 1948 [8]

2.3 TYPES OF ROBOTS

Generally, the types of robot are divided into two category in which:

1. Manipulator robotics (Robots stationed in one location and able to move other objects such as the robotic arm for material sorting process)

2. Mobile robotics (Robots which are able to move about)

2.3.1 Industrial Robots (manipulator robotics)

Typical industrial robots do jobs that are difficult, dangerous or which high repetition. They lift heavy objects, paint, chemical handling and perform assembly work. These robots are able to perform such task with high accuracy and precision. There are no downgrade or reduce quality of work due to fatigue or weariness and they are able to operate for long hours. Industrial robots used in the market include:

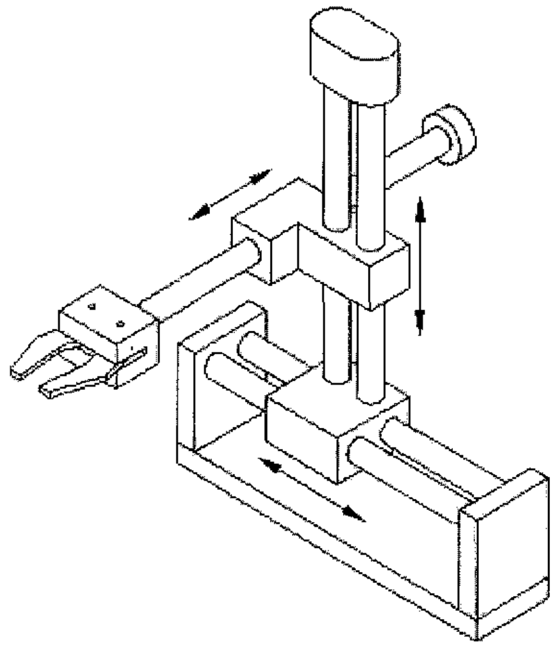


Figure 1 Diagram of a Cartesian Robot (XYZ robot)

Cartesian robot /Gantry robot:

Used for pick and place work, application of sealant, assembly operations, handling machine tools and arc welding. The robot is normally mounted on a tract on the operation floor. The Cartesian robot are the familiar of x,y and z axes of the machine tool.

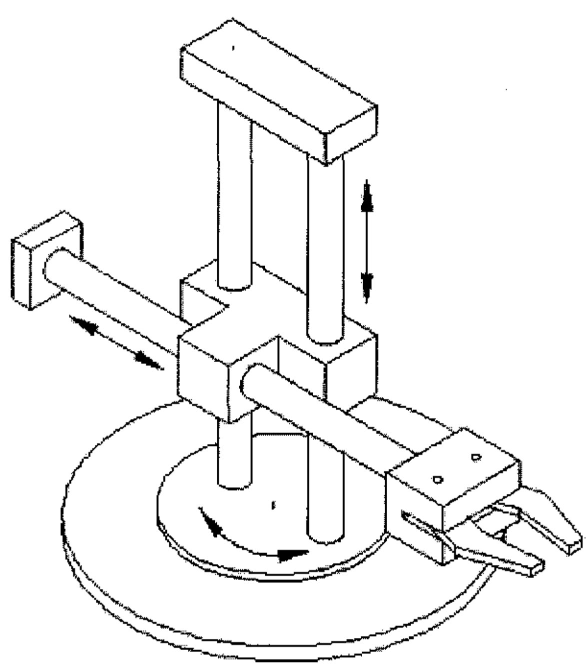


Figure 2 Diagram of a Cylindrical robot

Cylindrical robot:

Normally used for assembly operations, handling at machine tools, spot welding, and handling at diecasting machines. It's a robot whose axes form a cylindrical coordinate system. The movement of the joints of cylindrical robot generates a work shape of a cylinder.

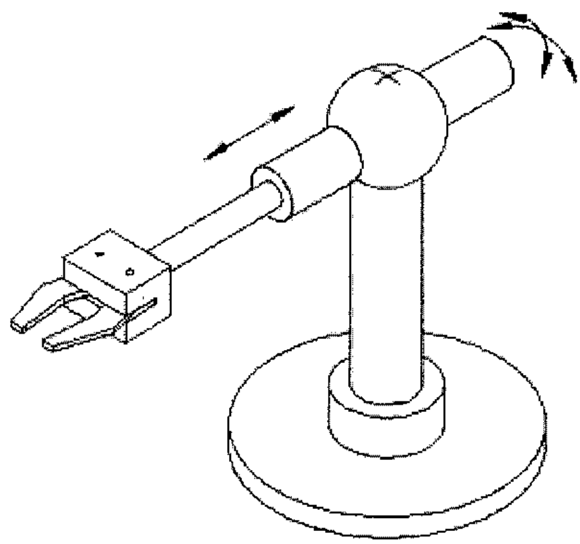


Figure 3 Diagram of a spherical robot

Spherical/Polar robot:

It is able to handle machine tools, spot welding, diecasting, gas welding and arc welding. It's a robot whose axes form a polar coordinate system.

SCARA robot:

Popular for pick and place work, application of sealant, assembly operations and handling machine tools. A robot which has two parallel rotary joints to provide compliance in a plane. The name SCARA means Selective Compliance Assembly Robot Arm which was introduced in the late 1970 as a robot ideally suited for assembly task. One of the main reason this robot excels is because of the Compliance Feature it offers. "Compliance" is a robotic term what means that the robot or tooling is capable of adjusting to accommodate misalignment. [3]

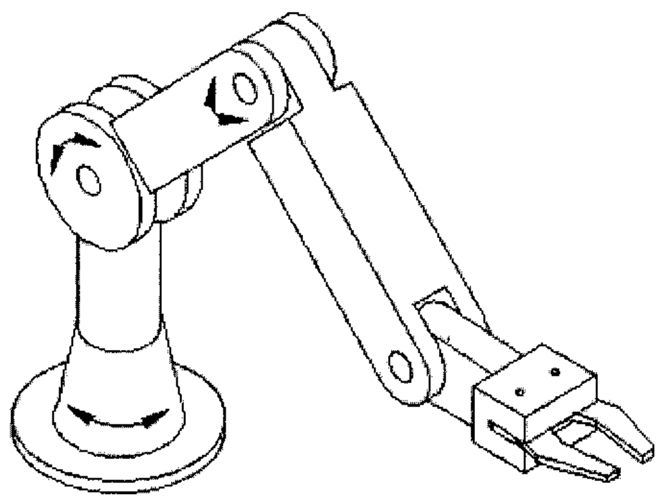


Figure 4 Diagram of an Articulated robot

Articulated robot:

A robot which resembles a robotic arm, used for assembly operations, diecasting, fettling machines, gas welding, arc welding and spray painting. It's a robot whose arm has at least three rotary joints.

Industrial robots are used in various fields such as automobile and manufacturing industries. Robots cut and shape fabricated parts, assemble machinery and inspect manufactured parts, carry heavy loads, perform spray painting and surface coatings and so on.

2.3.2 Mobile Robots

The other types of robots used are the mobile robots. These robots, unlike the industrial robots are able to move from one position to another. Such ability allows the robots to be used in various fields such as:

Replace human labour for long and tedious work (agriculture)

Farmers drive over a slow tractor for miles every year on the same ground. The slow speed and gentle land allows robot navigation techniques to be applied to this environment. Demeter, an automatic robot harvester, is a model for commercializing mobile robotics technology. The Demeter harvester contains controllers, positioners, safeguards, and task software specialized to the needs commercial agriculture.

Investigation of hazardous and dangerous environments

The Pioneer robot is a remote reconnaissance system, used to perform structural analysis of the Chornobyl Unit 4 reactor building. It is a teleoperated mobile robot for deploying sensor and sampling payloads, a mapper for creating photorealistic 3D models of the building interior, equipped with tools for cutting and retrieving samples of structural materials thus, providing an alternative to replace humans from entering hazardous environment. [7]

Dante II, an eight-legged, tethered, robot descended into the active crater of Mt. Spurr, an Alaskan volcano 90 miles west of Anchorage. The robot's mission was to walk autonomously over rough terrain in a harsh environment and determine the amount of carbon dioxide, hydrogen sulfide, and sulfur dioxide exist in the steamy gas emanating from fumaroles in the crater. With Dante II, many volcanologists are saved from having to enter the craters of active volcanoes. [7]

Underwater exploration

Robotic underwater rovers are used explore and gather information about many facets of our marine environment. Project Jeremy, collaboration between NASA and Santa Clara University. An underwater telepresence remotely operated vehicle (TROV) was sent by scientist into the freezing Arctic Ocean waters to investigate the remains of a whaling fleet lost in 1871. The TROV was cable operated, which carried power and instructions down to the robot and the robot returned video images. The TROV can also collect artifacts and gather information about the water conditions. [7]

Space Robots

Remotely Operated Vehicle (ROV) is used for outer space terrain exploration. An ROV can be of different forms , such as an unmanned spacecraft that remains in flight, or a rover that can move over terrain once it has landed. One of the best known ROV's is the Sojourner rover that was deployed by the Mars Pathfinder spacecraft. [7]

2.4 BUILDING BLOCKS OF ROBOTS AND AUTOMATION

Robots and automation is made up of various building blocks, in which are interdependent to each other for the successful implementation and function of the robot. Failure in one of these building blocks would affect the overall process of the automation. [3] The building blocks of a robot consist:

1. Controllers
2. Sensors and Transducers
3. Analyzers
4. Actuators
5. Drives

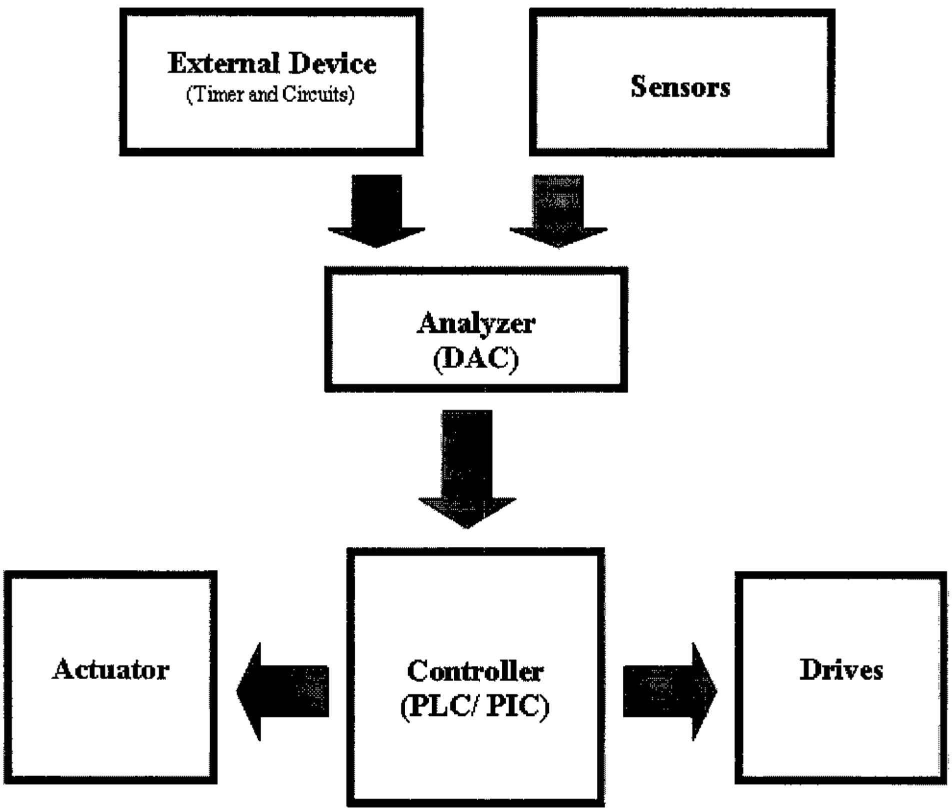


Figure 5 Building block components of robot and automations

2.4.1 Sensors

They are devices that generates output signal for the purpose of sensing a physical phenomenon. The most familiar sensor of all is a manual switch. The switch is the link between a robot and a person who desires the robot to be turned on or off. Limit switch is another type of mechanical switch. Unlike the manual switch, the input for a limit switch comes from the mechanism itself and is not controlled by the user. Limit switches can be used to limit the travel of a robot arm on any of its axes and motion. When the limit is reached, the circuit is opened that removes power from the axis of motion either directly or via the robot controller.

Another type of sensors are those that do not require physical contact. These proximity sensors are capable of sensing the presence of nearby objects without touching it. The proximity switches are such as the infrared sensors (uses infrared light to detect the presence of an object, not affected by ambient light), photoelectric sensors (uses light to detect presence of an object, affected by ambient light) and the ultrasonic sensors (uses sounds for detection; sound bounces back when hits solid barrier and detected by the transducer).

2.4.2 Controllers

These devices are known as the "brain" of the robot. It is able to define any action taken or to be perform according to the programme set by the programmer. The controllers' purposes are to monitor the coordination of the system such that a desired behaviour is achieved when an input is received.

Most common controllers used to control a robot are the PLC (Programmable Logic Controllers; normally used in large scale industries), Microcontrollers (such as the Programmable Input Controllers). Data and information will be inputted to the controllers, processed according to the programme defined and executed via the outputs.

2.4.3 Analyzers

The functions of analyzers are to register and analyse the output signals produced by sensors or devices. Analyzers for sensors may be of ADC (analogue to digital converter) in which is to convert data produced by the sensors to digital binary data understood by the robot.

2.4.4 Actuator

Once a real world condition is sensed and analyzed, something may be needed to be done. Actuators perform direct physical action for a particular process. The difference between actuators and drives is that actuators are limited to short linear discreet motion. Actuators may include solenoids and cylinders.

Cylinders are popular for pneumatic types and may be used in robots such as the gripper. The rod in the valve of the cylinder would extend when the air is pumped into the cylinder. The extended rod may be used as a switch to activate another push button, or just for mechanical joint extension used by pneumatic robot arms

2.4.5 Drives

Like actuators, drives take some action upon the process at the command of the controller or other analyzer. The difference between actuator and drives is that actuators are used to affect a short, complete, discrete motion (usually linear) and drives execute more continuous movements typically motors. Motors are the muscle of the robot; weather to move other objects or to move the robot itself. There are several types of motors normally used by robots, which are the stepper motors, DC continuous motors and the servo motors. Detailed discussions of these motors are explained in the report.

2.4.6 *Determining the Torque required by the motor*

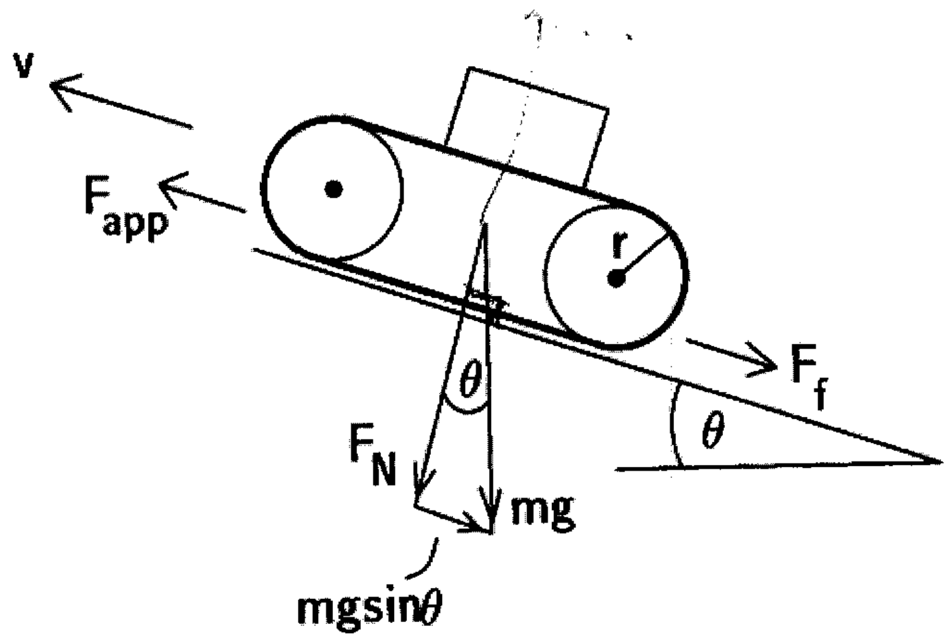


Figure 6 Free body diagram of a mobile robot and the forces as the vehicle climbs a hill.

In this project, it is assumed that the robot is mowing a flat land area. However, to overcome worst case scenario, the robot is required to climb a ramp of angle θ at a constant velocity, v.

$$F_{app} = F_f + F_w$$

Where Ff is the coefficient of friction, μ multiplied with the normal force, FN:

$$F_f = \mu F_N = \mu mg \cos\theta$$

Assumption:

m of robotic lawn mower = 5Kg

Coefficient of friction (tire on grass) μ = 0.35

The angle of robot climbing uphill, θ = 30 degree

Velocity of robot, v = 10cm/s

Diameter of wheel = 10cm

$$F_{app} = \mu mg\cos\theta + mg\sin\theta$$
$$F_{app} = (0.35)(5kg)(9.81m/s)\cos(30) + (5kg)(9.8m/s)\sin 30$$
$$F_{app} = 14.9695N + 24.525N$$
$$F_{app} = 39.4945N$$

The power required from the motor is the product of the force that needs to be applied by the wheels times the velocity, v the robot travel up the hill.

$$P_m = F_{app}v$$
$$P_m = 39.4945(0.1)$$
$$P_m = 3.949W$$

From the formula below, The Torque can be calculated

$$\omega = \frac{v}{r}$$
$$\omega = \frac{0.1m/s}{0.05m}$$
$$\omega = 2rad/s$$

$$\frac{P}{2} = T\omega$$

$$T = \frac{P}{2\omega}$$

$$T = \frac{3.949}{2(2)}$$

$$T = 0.98725Nm$$

Therefore, from the calculation, the minimum torque required is 0.98725Nm per motor for each wheel. Therefore, a motor which has a higher torque specification is chosen to support this robot, as the torque calculated does not include power loss of the motor and efficiency of the motor is not known.

2.4.7 Kinematics Motion using Differential Wheels

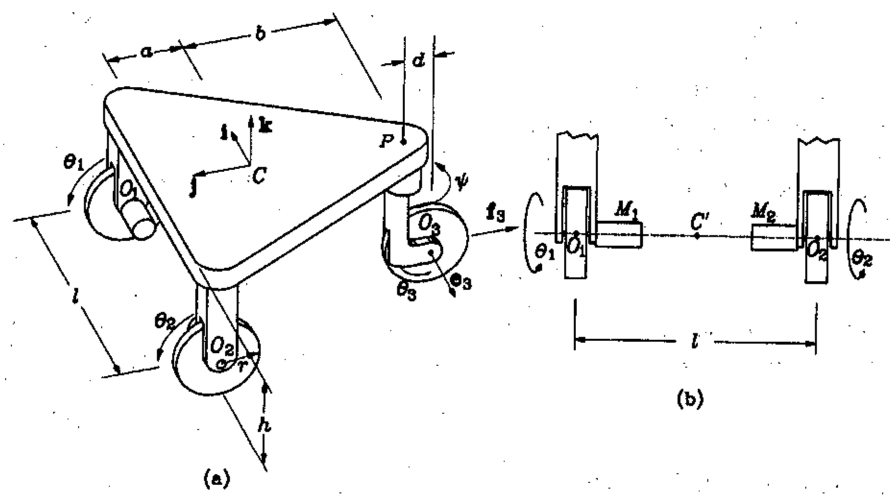


Figure 7 Mechanical variables of the robot (a) Top view of the robot. (b) the back view of the mobile robot(Picture taken from Fundamentals of Robotic Mechanical System by Jorge Angeles)

Scenario 1:

Robot is moving straight. Therefore, there is no relative change in the angular velocity of O_1 and O_2

$\theta 1 = \theta 2 = 2$ rad/s

Diameter, D = 10cm

Width of the robot (distance between 2 wheels) = 30cm

Center point = 2 cm from differential wheels (towards vector j)

Velocities $\dot{o}$ of point Oi, (for i = 1,2)

$$\dot{o} = r\dot{\theta}_i \mathbf{j}$$

$$\dot{o} = (0.05m)(2rad/s)_i \mathbf{j},$$

$$\dot{o} = 0.1m/s$$

And the angular velocity of ω of line O_1 and O_2 in planar motion, which is the same as that of the platform can be expressed by

$$\omega = \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)$$

$$\omega = \frac{0.05m}{0.3m}(2rad/s - 2rad/s)$$

$$\omega = 0rad/s$$

Therefore, there is no angle change in the motion of the robot, since both wheel are rotating at the same speed. Thus the robot is moving in a straight path.

$\dot{c}$ denotes the position vector of point C', the orthogonal projection of C onto the horizontal plane of O_1 and O_2,

$$\dot{c} = \alpha\frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)\mathbf{i} + \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2)\mathbf{j}$$

$$\dot{c} = 0.02m\frac{0.05m}{0.3m}(2rad/s - 2rad/s)\mathbf{i} + \frac{0.05m}{2}(2rad/s - 2rad/s)\mathbf{j}$$

$$\dot{c} = 0\mathbf{i} + 0.1\mathbf{j}$$

Therefore, the point $\dot{c}$ does not have any motion on the i-axis, since both wheels are turning at the same velocity. The robot is moving forward in the j-axis at 0.1m/s

Scenario 2:

Robot is performing a sharp turning to the right. Therefore, there is only angular velocity at O_1 and O_2 is not moving.

θ1 =2 rad/s

θ2 = 0 rad/s

Diameter, D = 10cm

Width of the robot (distance between 2 wheels) = 30cm

Center point = 2 cm from differential wheels (towards vector j)

Velocity $\dot{o}$ of point O_1 (since the robot is turning to the right, velocity of O_2 is 0 m/s)

$$\dot{o}_1 = r\dot{\theta}_i\,\mathbf{j}$$

$$\dot{o}_1 = (0.05m)(2rad/s)_i\,\mathbf{j},$$

$$\dot{o}_1 = 0.1m/s$$

And the angular velocity of ω of line O_1 and O_2 in planar motion, which is the same as that of the platform can be expressed by

$$\omega = \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)$$

$$\omega = \frac{0.05m}{0.3m}(2rad - 0rad)$$

$$\omega = 0.333rad/s$$

Therefore, the robot is rotating to the right with $\omega = 0.333rad/s$ with reference to the right wheel of the robot. The robot now moves in a circular motion with the right differential wheel as the pivoting point.

$\dot{c}$ denotes the position vector of point C', the orthogonal projection of C onto the horizontal plane of O_1 and O_2,

$$\dot{c} = \alpha\frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)\mathbf{i} + \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2)\mathbf{j}$$

$$\dot{c} = 0.02m\frac{0.05m}{0.3m}(2rad/s - 0rad/s)\mathbf{i} + \frac{0.05m}{2}(2rad/s + 0rad/s)\mathbf{j}$$

$$\dot{c} = 0.00666\mathbf{i} + 0.05\mathbf{j}$$

Therefore, the i-axis of point $\dot{c}$ increases by 0.00666 meters per second, since while moving forward in the j-axis at 0.05m/s

Scenario 3:

Robot is performing turning to the right. Both wheels are rotating at different speed with relative angular velocity of O_1 and O_2 is 2 rad/s

θ1 =4 rad/s

θ2 = 2 rad/s

Diameter, D = 10cm

Width of the robot (distance between 2 wheels) = 30cm

Center point = 2 cm from differential wheels (towards vector j)

Velocity $\dot{o}$ of point O_1 and O_2

$$\dot{o}_1 = r\dot{\theta}_i\,\mathbf{j}$$

$$\dot{o}_1 = (0.05m)(4rad/s)_i\,\mathbf{j},$$

$$\dot{o}_1 = 0.2m/s$$

$$\dot{o}_2 = r\dot{\theta}_i\,\mathbf{j}$$

$$\dot{o}_2 = (0.05m)(2rad/s)_i\,\mathbf{j}$$

$$\dot{o}_2 = 0.1m/s$$

And the angular velocity of ω of line O_1 and O_2 in planar motion, which is the same as that of the platform can be expressed by

$$\omega = \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)$$

$$\omega = \frac{0.05m}{0.3m}(4rad - 2rad)$$

$$\omega = 0.333rad/s$$

Therefore, the robot is rotating to the right with $\omega = 0.333rad/s$ with reference to the right wheel of the robot. The angle of rotation is the same as Scenario 2 since the relative difference of the speed between both wheels are 2 rad/s

$\dot{c}$ denotes the position vector of point C', the orthogonal projection of C onto the horizontal plane of O_1 and O_2,

$$\dot{c} = \alpha \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)\mathbf{i} + \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2)\mathbf{j}$$
$$\dot{c} = 0.02m \frac{0.05m}{0.3m}(4rad/s - 2rad/s)\mathbf{i} + \frac{0.05m}{2}(4rad/s + 2rad/s)\mathbf{j}$$
$$\dot{c} = 0.00666\mathbf{i} + 0.15\mathbf{j}$$

Therefore, the i-axis of point $\dot{c}$ increases by 0.00666 meters per second which is the same as Scenario 2. This is due that the relative difference in the velocity of the wheels are 2 rad/s (similar to scenario 2). However, the j-axis increases at 0.15m/s

2.4.8 Differential Wheels vs. Caster Wheel Mobile robot

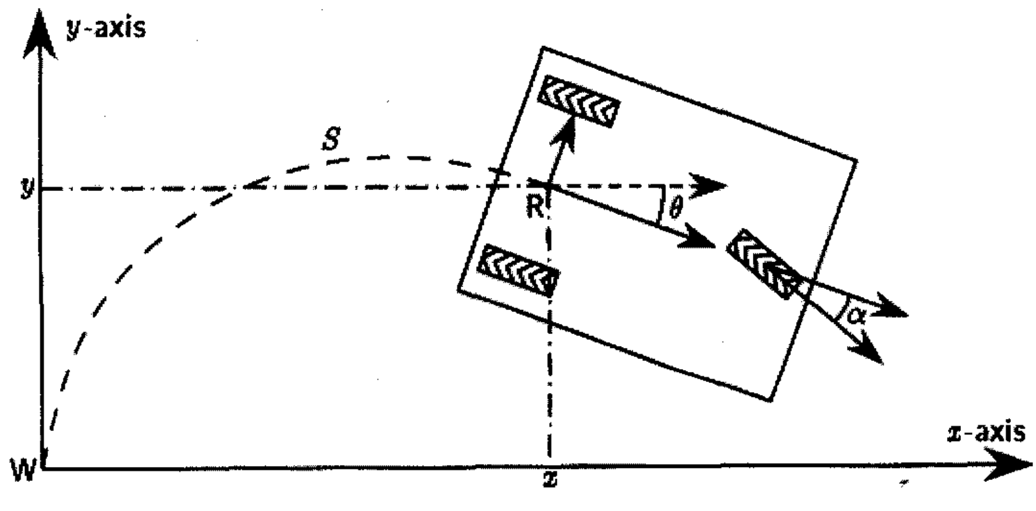


Figure 8 Kinematics of a Caster wheel drive mobile robot. The robot ahs three degrees of freedom in plane but only has two controllable parameters. Picture taken from MOBILE ROBOT: Inspiration to Implementation by Joseph L. Jones)

Differential drive robots have more advantage over caster wheel type and tricycle type robots. Relative to the global coordinate system, the robot can be position anywhere specified by 2 coordinates of x and y and pointed in any direction specified by a third coordinate, angle θ. [4]

Robots with Caster wheels however, the degree of freedom of the robot is limited to two parameters; the steering angle, α and the total distance it travels, S. Therefore, the robot's orientation and position are coupled. The robot must move forward or backward in order to turn. The robot can not go directly from one position/orientation to another. The robot must follow some path and would be complicated with the presence of obstacle.

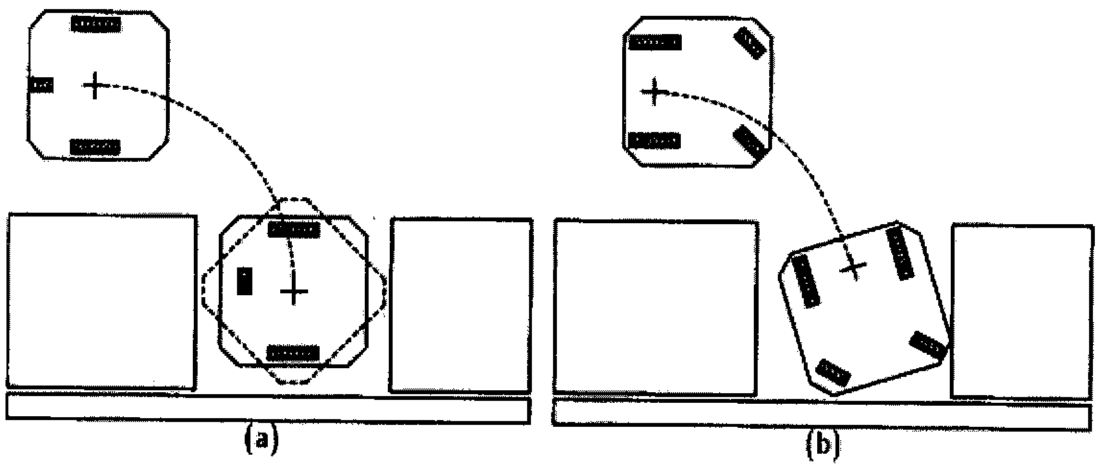


Figure 9 Comparison of differential drive and caster wheel drive kinematics using parallel parking example. (a) Differential drive robot has more flexibility in reaching the desired position and orientation. (b) Caster Wheel drive requires turning forward and backward motions in order to reach desired position and orientation. (Picture taken from MOBILE ROBOT: Inspiration to Implementation by Joseph L. Jones)

The differential drive method is chosen for this robot as it is easier to implement and the amount of flexibility is much more as compared to the mobile robot with caster wheel drive.

2.5 CONTROLLERS

2.5.1 Programmable Logic Controller (PLC)

The Programmable Logic Controller is a digitally operating electronic apparatus which uses a programmable memory for the internal storage of instructions by implementing specific functions such as logic sequencing, timing, counting, and arithmetic to control, through digital or analogue input/output modules

2.5.2 PIC Microcontroller

The Programmable Input Controller (PIC) microchips are used for embedded designs. The microcontroller is able to perform various functions according to the user programming codes (Assembly Language/ C Programming). The microcontroller is able to perform task which is once difficult to be implemented on normal logic circuit.

Table 1 Comparison between Wired Logic, PLC and PIC controllers

	Wired Logic	PLC	PCI
Control device (Hardware)	Specific purpose	General purpose	Specific purpose but programmable
Control Scale	Small and medium	Medium and large	Small scale is likely to succeed
Change or addition of specification	Difficult	Easy	Moderate (depends on the programming skill of user)
Delivery period	Several days	Almost immediate	Almost immediate
Maintenance (by makers and users)	Difficult	Easy	Moderate
Reliability	Depend on design and manufacturer	Very high	Not able to withstand industrial environment
Economic efficiency	Advantage on small scale operation	Advantage on small, medium and large scale scale operation	Cheap

2.6 ROBOTIC FUNDAMENTAL

2.6.1 Fundamental of Robotic Mechanical System

Rolling robots are under the autonomous mobile robot category. The focus is on robots meant for tasks on horizontal surface, in which their platforms undergo planar motion, which simplifies their kinematics. Rolling robots are basically of two kinds, depending on whether they are supplied with conventional or omni directional wheels. Robots with conventional wheels have 2 degrees of freedom motion while robots with omnidirectional wheels are capable of 3 degrees of freedom motion which increase their maneuverability substantially

2.6.2 Robots with Conventional Wheel

These robot have only 2 degree of freedom, therefore, they only need two actuators. The example below is taken from *Fundamentals of Robotic Mechanical Systems (Theory, Methods and Algorithms) written by Jorge Angeles.* The basic architecture of this type of robot is shown in figure 1, which distinguish a chassis (robot body). Two coaxial wheels are coupled to the chassis and revolute of axes passing through points O1 and O2. [5]

The orientation of the steering wheel is defined by ψ, is controlled by the second actuator. The design has a few drawbacks:

1. Two motors serving two essentially different task requires different operational characteristics, in which both may not be available from the same manufacturer.
2. Power motor requires velocity control, steering motor for position control therefore giving rise to two independent control systems that may end up by operating in uncoordinated fashion
3. Differential gear train increase cost, weight and bring out inherent backlash of gears.

2.6.3 Analysis of kinematics of a robot with differential wheels

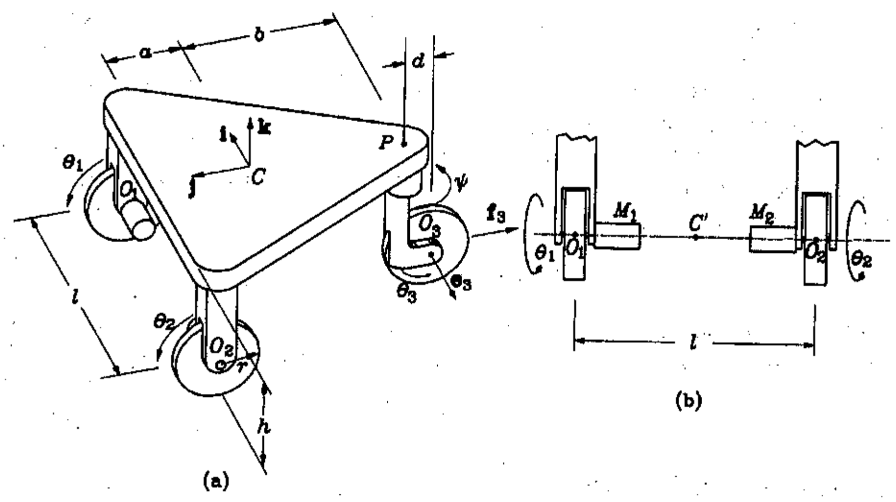


Figure 10 Mechanical variables of the robot (a) Top view of the robot. (b) the back view of the mobile robot

Point C of the platform is taken to be the **operation point**; its position vector in a frame fixed to the ground denoted by ***c***. ω is taken to be the scalar angular velocity of the platform about a vertical axis. By virtue of the 2 degree of freedom of this robot, the velocity $\dot{\mathbf{c}}$ of C can be controlled or a combination of ω and a scalar function of $\dot{\mathbf{c}}$ by properly specifying the two joint rate Θ_1 and Θ_2. However, the two components of $\dot{\mathbf{c}}$ and ω cannot be controlled simultaneously. [5]

An orthonormal triad of vectors whose orientation is fixed with respect to the chassis. The triad is denoted by {**i, j, k**} with k pointing upward vertical direction. [5]

Therefore, the velocities $\dot{o}$ of point Oi, (for i = 1,2)

$$\dot{o} = r\dot{\theta}_i\, j, \quad \text{for } i = 1,2$$

And the angular velocity of ω of line O_1 and O_2 in planar motion, which is the same as that of the platform can be expressed by

$$\omega = \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)$$

Furthermore, the velocity of C can now b written in 2 dimensional from as

$$\dot{c} = \dot{o}_1 + \omega E(c' - o_1)$$

With $\dot{c}$ denoting the position vector of point C', the orthogonal projection of C onto the horizontal plane of O_1 and O_2, while E as an orthogonal matrix rotating 2 dimensional vectors through an angle of 90 degree counterclockwise. [5] Hence,

$$E = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Thus, all vectors are 2 dimensional. Upon substitution of equation $\omega = \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)$ with $\dot{c} = \dot{o}_1 + \omega E(c' - o_1)$. We obtain an expression for $\dot{c}$ in terms of the joint rates, namely,

$$\dot{c} = \alpha \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2) i + \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2) j$$

2.7 MECHANICAL DRIVES

2.7.1 Choosing the Right Motor for the Job

Motors are the muscle of robots, when attached to a set of wheels, robots are used as legs. When attaching a motor to a lever, they serve the purpose as a shoulder joint to move up and down. There are many types of motors:

DC motor: The common motor normally used to drive continuously in one direction. It only stops when the power supply is removed.

Stepper motor: Application of power causes the shaft to rotate a few degrees and then stop. Continuous rotation of the shaft requires the power to be pulsed to the motor.

Servo Motors: Strong motors normally used as joints for robots. It can only move 180 degrees in direction

2.7.2 Motor Specification

Voltage

All motors are rated by their operating voltage. Small hobby motors usually range from 1.5 V to 12 V. Most motor can be operated satisfactorily at voltages higher or lower than that specified. However, most motor is likely to operate 50 percent of the specified rating. Running a motor continuously at more than 30 or 40 percent of its rated voltage is not recommended. The windings will overheat which may cause permanent damage.

Current Draw

It is the amount of current (amp or milliamps) in which the motor requires from the power supply. Current draw however, increases with load. A point is reached when the motor does all the work it can perform, and no more current will flow through it. The shaft strops rotating; the motor has stalled. Therefore, when building the robot, the load (weight) must be including under consideration.

Speed

The rotational speed of a motor is given in revolution per minute (rpm). The speed of the motor could be increase or decreased using 2 methods: building a bigger motor (impractical) or add gear reduction.

The speed always decrease when going from small to large gear

The speed always increases when going from large to small gear

Therefore, if there is a need to reduce the speed of motor from 5000 rpm to 50 rpm, The speed reduction would require a reduction ratio of 100:1. This would require a drive gear of 10 teeth and a driven gear of 1000 teeth.

2.7.3 Torque

Torque is the angular force that a motor can deliver at a certain distance from the shaft. A 5 oz-in of torque means that at a distance of 1 inch away from the shaft of the motor, it is strong enough to pull up a weight of 5 ounces using a pulley. Metric unit normally specify torque in terms of Newton-meters (Nm). [4]

$$1N = 1\frac{kg-m}{\sec^2} = 0.225lb$$

Electrical power is converted to mechanical power in a motor. The relationship involves power (in watts) and energy (in joules). Power is the rate of energy used. Therefore, power is represented by

$$1\ Watt = 1\frac{Joule}{\sec}$$

The electrical power supplied to the motor, Pe equals the voltage, V across the motor's terminal multiply with the current, I through the motor. Current, measured in the unit of amperes is the amount of charge passing through a conductor per second: [4]

$$P_e = VI$$

$$1\ Ampere = 1\frac{Coulomb}{Sec}$$

$$1\ Watt = 1\ Volt * Ampere = 1\ Volt * \frac{Coulomb}{\sec}$$

Mechanical power, Pm equals the torque, T output by the shaft multiplied with its angular speed, ω (torque is taken in Newton Meters):

$$P_m = T\omega$$

$$\frac{2\pi rad}{\sec} = 1\frac{rev}{\sec}$$

$$1\ Watt = 1\ \frac{Nm}{\sec}$$

Since power is energy per unit time, this shows that one joule energy can be expressed in two ways; either as 1 Newton meter or 1 Coulomb Volt. The is just reaffirming the fact that energy is energy whether it comes from mechanical origin or electrical origin. [4] The motor is just a transducer transforming energy to one form from another.

$$1\ \mathbf{J} = 1\ Nm \qquad\qquad 1\ \mathbf{J} = 1\ CV$$

2.7.4 Motor Model

The relationship describing the electrical power to mechanical power in permanent DC motor is described previously. The mechanical output power (due to the losses from friction, windage, heating in the coils) will be some fraction of the electrical input power. [4] The percentage is given as efficiency, η where:

$$P_m = \eta P_e$$

The rotor coil is essentially an inductor with a resistance R. When rotor is turning, the commutator segment sliding past the brushes create an alternating current in the armature winding. A changing current, $\frac{di}{dt}$ through an inductor induces a voltage across it:

$$v = \mathbf{L}\frac{di}{dt}$$

Where L is the proportionality constant called the inductance. As the motor turns voltage is induced and opposes the applied voltage. The faster the motor turns, the more the current switches direction, thus the larger the induced voltage becomes. Thus, it tends to limit the current through the resistor R.

When current falls, the less flux is created around the conductor, and the torque also falls. Therefore, when speed increases, the torque decreases.

The rotating motor can e modeled by the induced voltage, e (back electro magnetic force) and the winding resistance R. The applied voltage is related to the back--emf and current:

$$V = IR + e$$

when motor is not rotating, e = 0V and the current through the motor is equal to the applied drive voltage divided by the resistance. The current required to start the motor from zero speed is called the starting current/stall current, Is.

$$I_s = \frac{V}{R}$$

When the rotor is rotating, e increases proportionally with the speed of the armature:

$$e = k_e \omega$$

where ke is called the back-emf constant. The applied voltage is then related to the current and the armature speed by:

$$V = IR + k_e \omega$$

The negative feedback provided by the back-emf causes the motor to settle to a steady state operating point of speed and torque as determined by the load and the applied voltage. The torque that the motor produce is dependent on the flux is controlled only by the current. The torque increases linearly with current with proportionality constant kt (torque constant):

$$T = k_t I$$

solving for I and plugging into the equation above:

$$V = \frac{TR}{k_t} + k_e \omega$$

kt is actually equal ke. This can be seen from the fact that the mechanical power output by the shaft will be the electrical power input, minus the $I^2 R$ losses due to heating in the resistor:

$$P_m = P_e - I^2 R$$
$$T\omega = VI - I^2 R$$

Replacing the equation for T and V

$$k_t I \omega = (IR + k_e \omega) I - I^2 R$$

gives,

$$k_t = k_e = k$$

The applied voltage is then related to the torque and speed by constant k:

$$V = \frac{TR}{k} + k\ \omega$$

the speed-torque relationship is linear with a negative slope:

$$\omega = -\frac{R}{k^2}T + \frac{V}{k}$$

2.7.5 Determining the amount of power needed for the motor

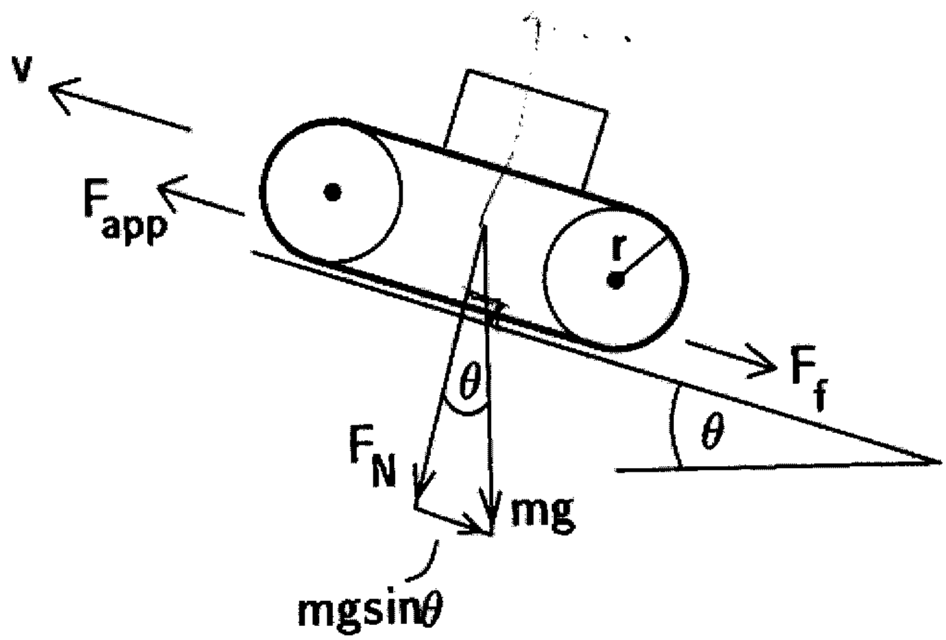


Figure 11 Free body diagram of a mobile robot and the forces as the vehicle climbs a hill (picture taken from MOBILE ROBOT: Inspiration to Implementation by Joseph L. Jones)

Assuming that the mobile vehicle is a differential drive mechanism (two motors) and need to climb a ramp of angle θ at a constant velocity v. The free-body diagram makes explicit the force acting on the vehicle. [4]

Since the vehicle moves at a constant velocity, there are no net force on the car:

$$F = ma$$

Since the acceleration a, is 0 (the car moves at a constant velocity), The net force F must be 0. Thus, the applied force F_{app}, from the wheels acting in the direction up the hill must balance the force down the hill resisting that force. The resisting force are the friction force and the force that is the component of the vehicle's weight acting in the direction down the hill. [4]

$$F_{app} = F_f + F_w$$

Where F_f is the coefficient of friction, μ multiplied with the normal force, FN:

$$F_f = \mu F_N = \mu mg\cos\theta$$

F_w is the mg sin θ (where mg, mass multiplied with acceleration due to gravity is the weight of the robot)

$$F_{app} = \mu mg\cos\theta + mg\sin\theta$$

The power required from the motor is the product of the force that needs to be applied by the wheels times the velocity, v the robot travel up the hill.

$$P_m = F_{app} v$$

The torque and speed requirement of each motor can be calculated from:

$$\frac{P}{2} = T\omega \qquad\qquad \omega = \frac{v}{r}$$

2.7.6 Robot Locomotion with DC Motors

Most robot design use two identical motor to spin two wheels. These wheels provide forward and backward momentum, as well as left and right steering. Stopping one motor, would allow the robot to change direction. Reversing both motors in relative to one another, robot turns by spinning on its wheel axis. This would cause sharp right and left turns.

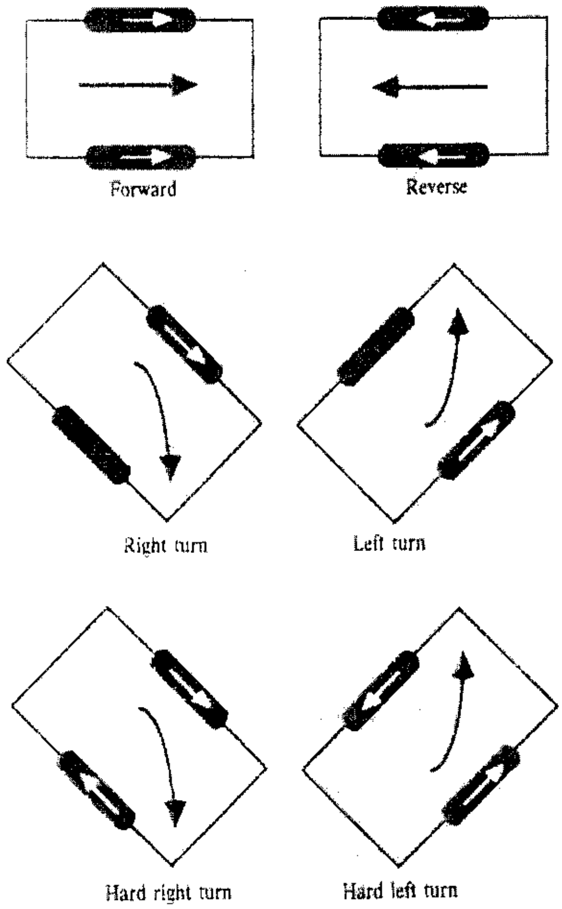


Figure 12 Directions of mobile robot taken from Gordon Mc Comb: Robot Builder's Bonanza

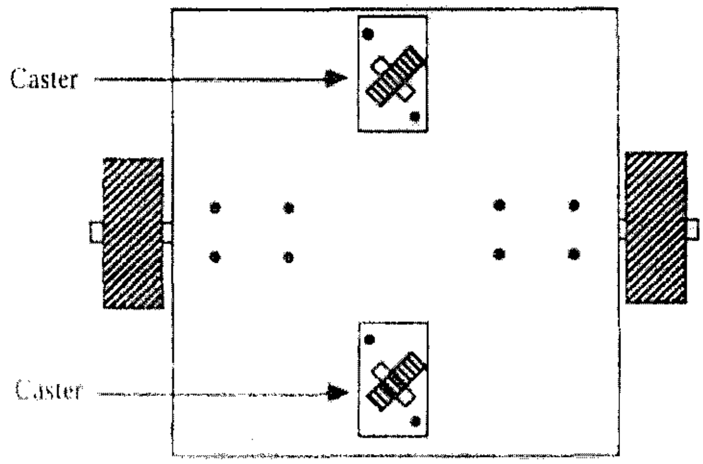


Figure 13 A two wheeled mobile robot taken from Gordon Mc Comb: Robot Builder's Bonanza

Method for calculating travel speed of the robot:

1. Divide the rpm speed of the motor by 60. The result is the revolutions of the motor per second (rps). Example: A 100 rpm motor runs at 1.66 rps.
2. Multiply the diameter of the drive wheel by pi ($\pi = 3.142$). Example: A 5 cm diameter wheel would has a circumference of about 15.71 cm.
3. Multiply the speed of the motor (rps) with the circumference of the wheel. The result would be the number of centimetres covered by the wheel in one second.
4. The heavier the robot, the slower the motor will turn (depending on the torque).

2.7.7 Robot Locomotion with Stepper Motors

Stepper Motors

DC motors are cheap and able to deliver a lot of torque for their size. However, the common dc motor is rather imprecise because there is not servo feedback mechanism or tachometer. It is impossible to command the motor to turn exactly a specific number of revolutions and also fraction of a revolution. Stepper motors are dc motor with slight difference: instead of being powered by a continuous flow of current, as with regular dc motors, they are driven by pulses of electricity. Each pulse drives the shaft of the motor a little bit. The more pulse that are fed to the motor, the more the shaft turns. [8]

Figure 14 Picture of a stepper motor taken from Gordon Mc Comb: Robot Builder's Bonanza

Inside a stepper motor

A four phase stepper motor is really two motors switched together. Each motor is composed of two windings. Wire connects to each of the four windings of the motor pair, so there are eight wires coming from the motor. The common from the windings are often ganged together, which reduces the wire count to five or six instead of eight.

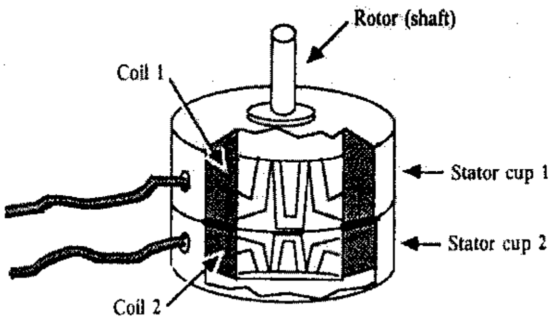


Figure 15 Illustration of a Stepper motor and the components that make up a stepper motor

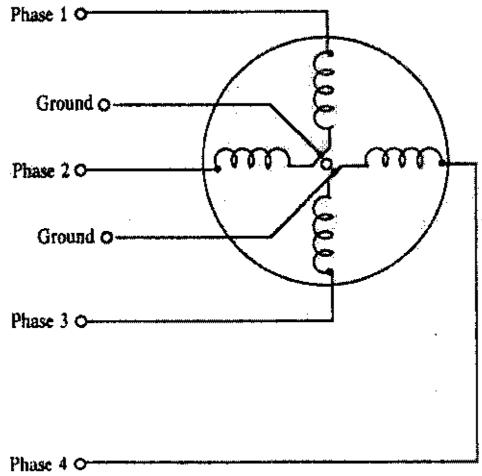


Figure 16 Schematic diagram of the Stepper motor

Wave Step Sequence

The motor shaft turns a fraction of a revolution each time a winding is energized. For the shaft to turn properly the winding must be energized in wave step sequence (energized wires 1, 2, 3 and 4)

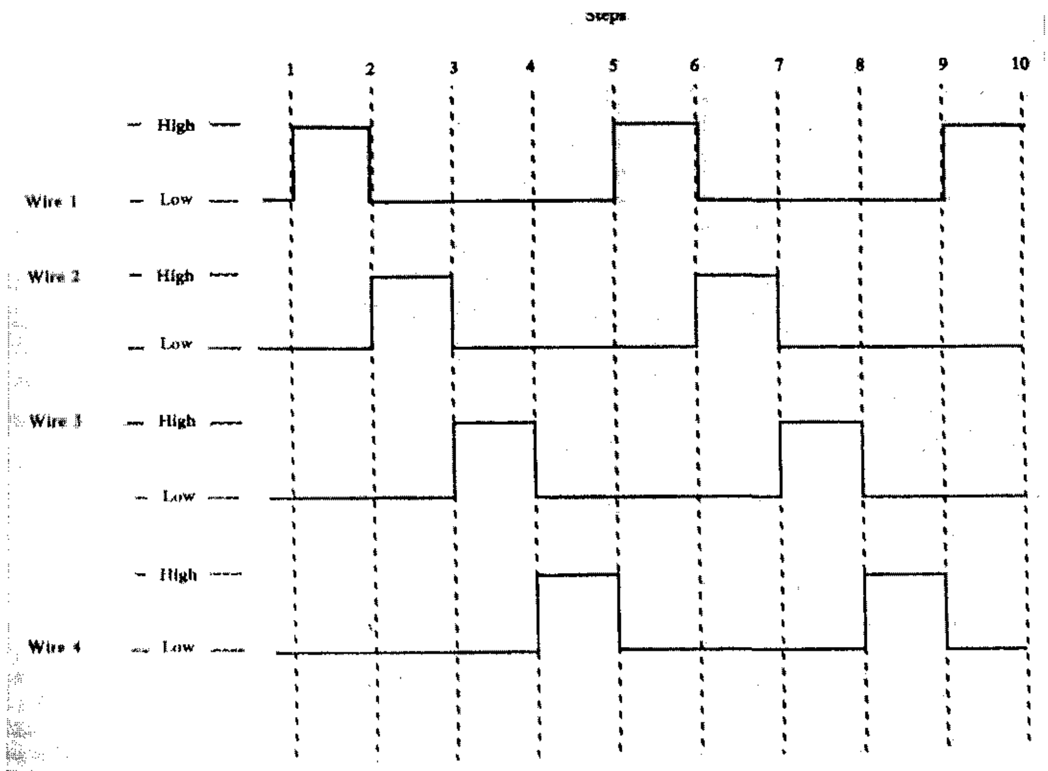


Figure 17 Basic wave-step actuation sequence of four-phase stepper motor

Double-On/Double-Off Sequence

Wave step sequence is the basic actuation technique for a four-phase motor. A better approach would require actuating two winding at one in both-on/ both-off sequence increasing the driving power of the motor. This provides greater shaft rotation precision. Other varieties of stepper motors would require actuating it in different ways.

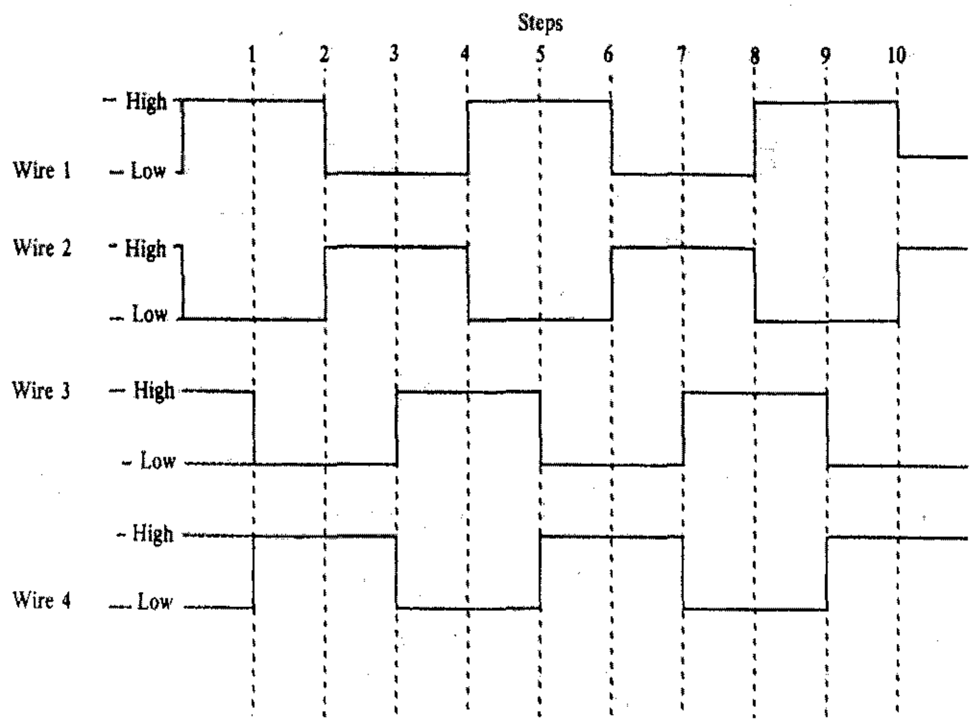


Figure 18 The double-on/double-off actuation sequence of four-phase stepper motor

2.7.8 Stepper Motor Specification

Stepper Phasing

A four-phase stepper requires a sequence of four pulses applied to its various windings for proper rotation. Most stepper are at least 2 phase, however majority are 4 phase and six phase. Usually the more phases in a motor, the more accurate they are.

Step Angle

Stepper motors vary in amount of rotation the shaft turns each time a winding is energized. The amount of rotation us caked "step angle" and can vary from as small as 0.9 degrees (1.8 degree is more common) to 90 degrees. The step angle determines the number of steps per revolution. A stepper with a 1.8 degree step angle must be pulsed 200 times for the shaft to turn one complete revolution. A stepper with a 7.5 degree step angle instead must be pulsed 48 times for one revolution.

Pulse rate

The smaller the step angle, the more accurate the motor. But stepper motor has an upper limit to the number of pulses they can accept per second. Heavy duty stepper usually have maximum pulse rate of 200 to 300 steps per second providing 60 to 180 rpm. Smaller stepper can accept thousand or more pulses per second but they do not provide much torque and are not suitable as driving or steering motors.

An interesting point to note is that stepper motor can not be motivated to run at their top speed immediately from a dead stop. Applying too many pulses directly from the battery causes the motor to freeze up. To achieve top speed, the motor must be gradually accelerated. In terms of human motion, the acceleration can be quite swift. The speed can be 1/3 for the first milliseconds, 2/3 on the next 50 to 75 miliseconds and full speed later on.

Running Torque

Steppers may not be able to deliver as much torque as standard dc motors for the same size and weight. In comparison, a typical 12 volt medium size stepper motor may have torque of 25 oz-inches. The same 12 volt, medium size standard dc motor may have running torque tree or four times more.

Steppers are at their optimum performance when turning slowly. With stepper, the slower the motor revolves, the higher the torque. DC motor however, is opposite from this situation, in which the higher the speed, the better the torque.

Breaking Effect

Actuation of one of the winding in a stepper motor advances the shaft. Continuous input of current to the winding and the motor would not turn anymore. The shaft would be locked as if the breaks were applied on to them.

2.8 ELECTRONIC CIRCUITS

2.8.1 Collision Avoidance and Detection

These are the two different types of detection system. Collision Detection is a form of Passive detection, the robot only changes direction after it has banged into an obstacle along its moving path. Collision Avoidance however, is more towards **Active** detection system. The system would detect an obstacle from a defined distance (dependant on the sensitivity of the sensors), and changes course of direction before collision.

Infrared Light

Light may always travel in a straight line but it bounces off nearly everything. This is advantageous to build an infrared collision detection system. The circuit below is an example on how the infrared LED and phototransistors can be mounted on the top of the robot for the purpose of detecting an obstacle like a wall and rock. The set point adjustment, R2, provides a means to increase or decrease sensitivity of the circuit. Increase in sensitivity would allow the robot able to detect objects future away. The circuit recommended below has an effective range of 6 inches.

Objects reflect light in different ways. Light bounces better on white coloured objects compared to black. The phototransistor must be blocked from the direct light of the LED.

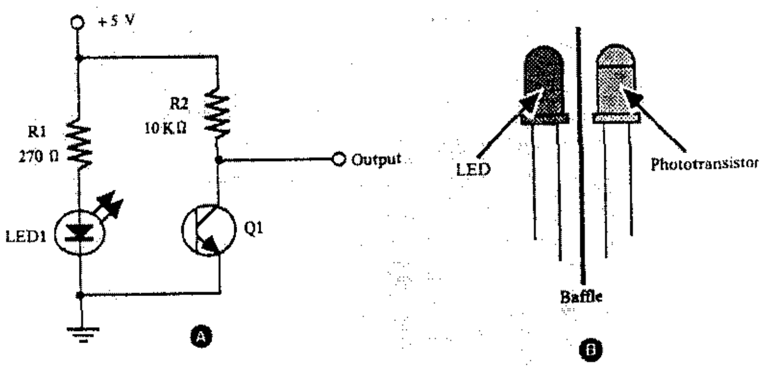


Figure 19 Proximity detector using infrared light circuit diagram; b: LED/Phototransistor Placement

Ultrasonic Sound

Sound can be used o detect the proximity of objects in much the same as for infrared light. Ultrasonic sound is transmitted from a transducer, reflected by a nearby object, and then received by another transducer. The advantage of using sound is that it is not sensitive to objects of different colour and light reflective properties. However, there are materials reflect sound better than others, and some even absorb sound completely. In comparison, proximity detection with sound is more fool proof.

The circuit below provides a practical circuit for building ultrasonic proximity detector. A stream of 40 kHz pulses is produced by a 555 timer wired up as an astable multivibrator. The output of the 555 provides more than enough power for the transducer. A piece of foam between the two transducers is needed to eliminate direct interference between the two.

The advantage of using ultrasonic is that it is particular about the frequency of sound. Specifically, it is desired to limit the sensitivity of the circuit to 40 kHz, the same as the output of the transmitter. A 567 tone decoder IC is connected to the output of the 741 amp.

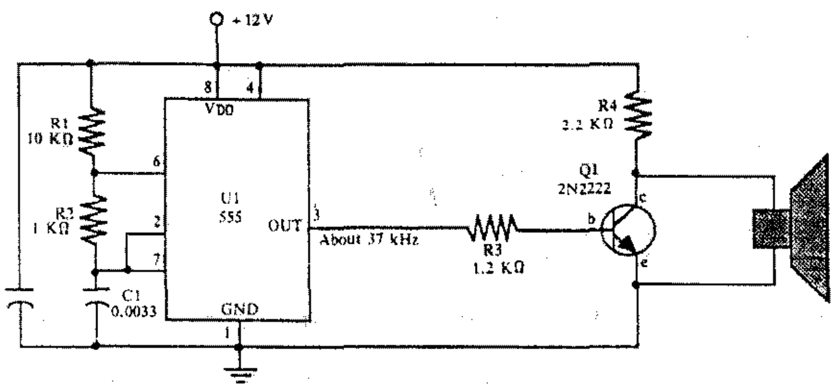


Figure 20 Schematic of 40 kHz ultrasonic transmitter (taken from Robot Builder's Bonanza by Gordon McComb)

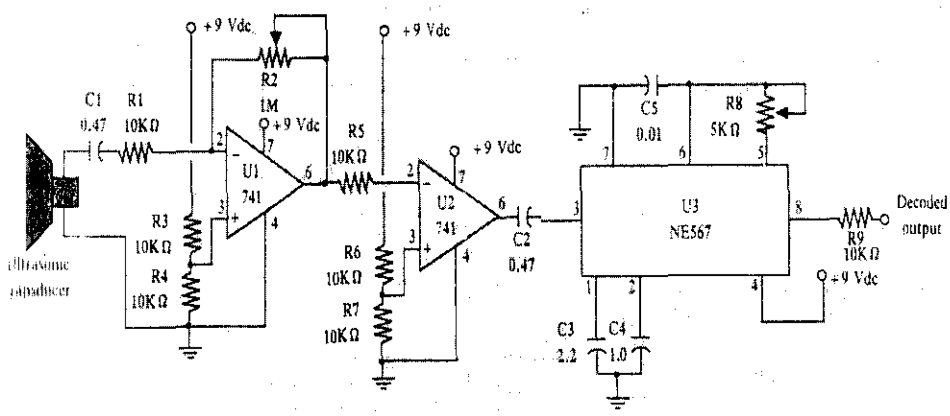


Figure 21 Schematic for an ultrasonic receiver and tone decoder (taken from Robot Builder's Bonanza by Gordon McComb)

2.8.2 Direction Control

Relay Control

The method is slightly old fashion compared to the other alternatives. However, they are less expensive than the other methods, easier to implement and take up less space. The relay may wear out in time (after a few hundred thousand switching)

This direction control system however, does produce a centre-off position, in which the motor receive no power and does not move. The diagram below shows a simple motor switching circuit. Input to the relay would trigger both the motor to operate.

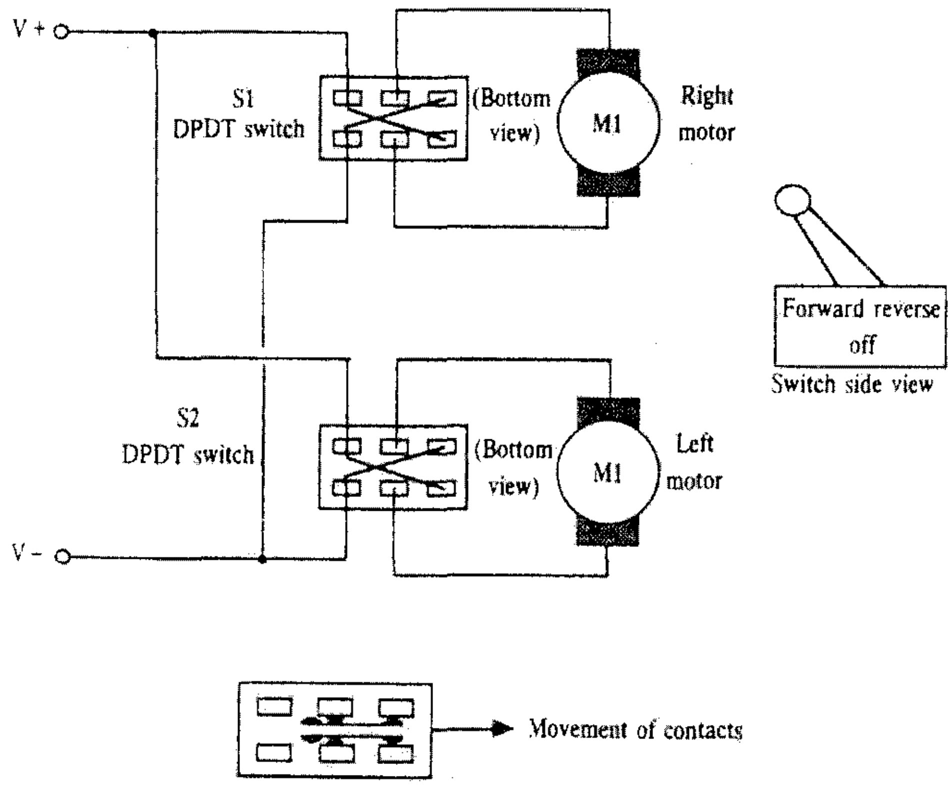


Figure 22 A relay controlled motor circuit taken from Gordon Mc Comb: Robot Builder's Bonanza

Transistor

Transistors provide true solid state control of motors. The motor is connected so that when one transistor is switched on, the shaft turns clockwise. When the other transistor turns on, the shaft turns counter-clockwise. When both transistors are off, the motor does not turn. The amount of voltage requires depends on the volt the motor requires. However, for this motor control, both motors can not be turned on at the same time. Doing so will cause damage to the transistors.

Resistors used to bias the base of each transistor are necessary to prevent the transistors from pulling excessive current from the gate controlling it (computer port, microcontroller or logic gates) Without resistor; the gate would overheat and be destroyed. The actual value of the resistor depends on the voltage and current draw of the motor, as well as the characteristic of the transistors used.

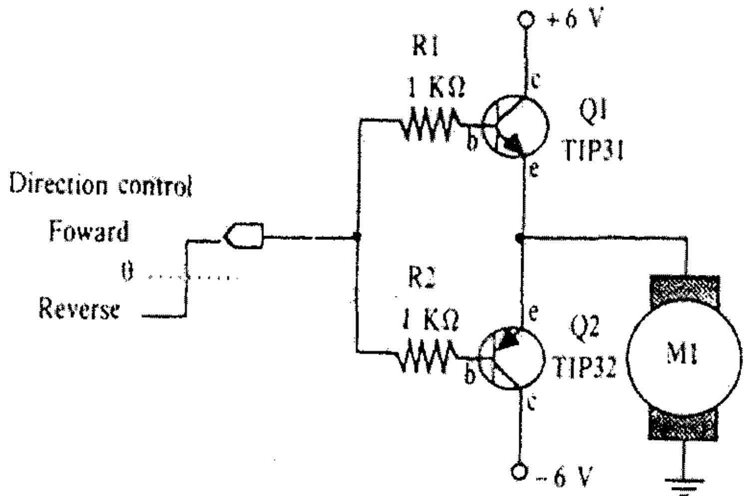


Figure 23 A transistor controlled motor circuit taken from Gordon Mc Comb: Robot Builder's Bonanza

The diagram below shows the "H" network which is wired such a way that only two resistors are on at a time. When transistor 1 and 4 are activated, the motor turns in one direction. When transistor 2 and 3 are on, the motor spins the other way. When all transistors are off, the motor remains still.

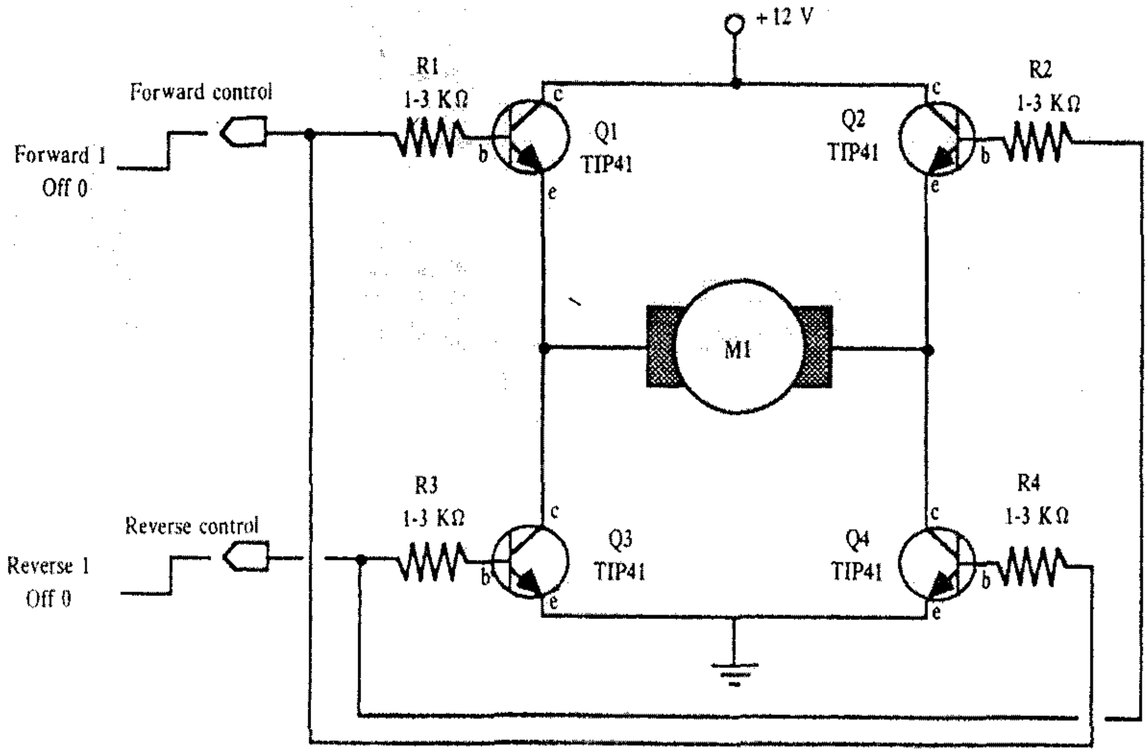


Figure 24 An "H-Bridge" motor control circuit

There is a special type of transistor, which is the power MOSFET (Metal Oxide Semiconductor Field Transistor). The power MOSFET is able to handle higher current and voltage to drive the motors without worrying about burning and frying the component.

The diagram below uses power MOSFET to drive the motor system. The circuit is controlled by a NAND CMOS gate for positive action control. [8]

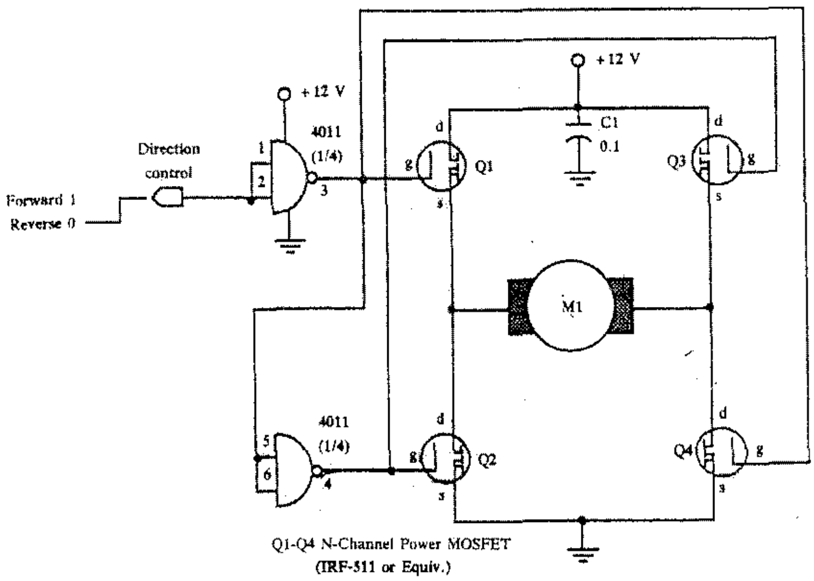


Figure 25 Diagram of a power MOSFET and "H-Bridge" motor control circuit using Power MOSFET (taken from Gordon Mc Comb: Robot Builder's Bonanza)

Stepper Motor Control Circuit

The stepper motor requires a controller or translator circuit since the movement of the stepper motors are determined by the phase inputted to the motor. The circuit can be constructed using TTL or CMOS chips. The 7486 (exclusive OR) is used to provide steering logic. The steeping actuation is controlled by 7476 (JK flip flop) The output of Q and Q' produces control to the phasing of the motor. Stepping is accomplished by triggering the clock input of both flip flop. [8]

The 7476 is unable to power a stepper motor directly. An external voltage supply is needed to provide power to the stepper motor. The input to the phase is controlled by npn transistor. The transistors shown in the diagram is 2N3055. Since the transistors are TTL chips, biasing resistors are needed to produce current to the input base of the chips. Resistors ranging between 1K Ohms to 3 K Ohm. [8]

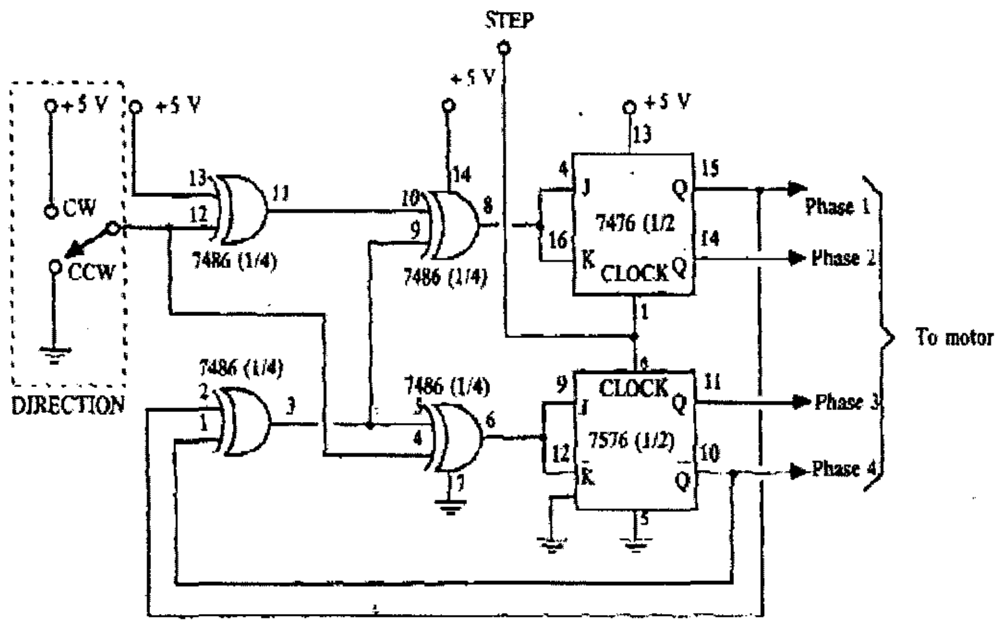


Figure 26 Stepper Motor translator circuit using the 7486 (exclusive OR) chip and 7476 (JK flip flop) chip. taken from Gordon Mc Comb: Robot Builder's Bonanza

CHAPTER 3
METHODOLOGY / PROJECT WORK

3.1 TOOLS REQUIRED

I. Programmable Input Controller (PIC) 16F84A and PIC 16F877
II. PIC Programmer Board
III. PIC 16F84 and PIC 16F877 Target boards
IV. DC motors (x2)
V. Ultrasonic Sensors sensors (x3)
VI. Plastic wheels and Gear ratio mechanism (a pair)
VII. Caster wheels
VIII. L shape aluminium bars (for chasis construction)
IX. Rechargeable battery (vision 12V battery)
X. Screws and nuts
XI. Personal Computer (for C++ programming)
XII. A 12 Volt RS Stepper Motor
XIII. Stepper Motor Controller
XIV. H-bridge circuits
XV. Softwares:
    a. CCS-Compiler
    b. Electronic Workbench
    c. Bumble Bee Software (for interfacing with the programmer board)

Equipments and tools needed for this project are:

ii) Personal Computer
iii) Programmable Input Controller (PIC)
iv) PIC Programmer
v) Sensors, Motors and relevant digital electronic components to develop the circuitry

3.2 PROCEDURE IDENTIFICATION

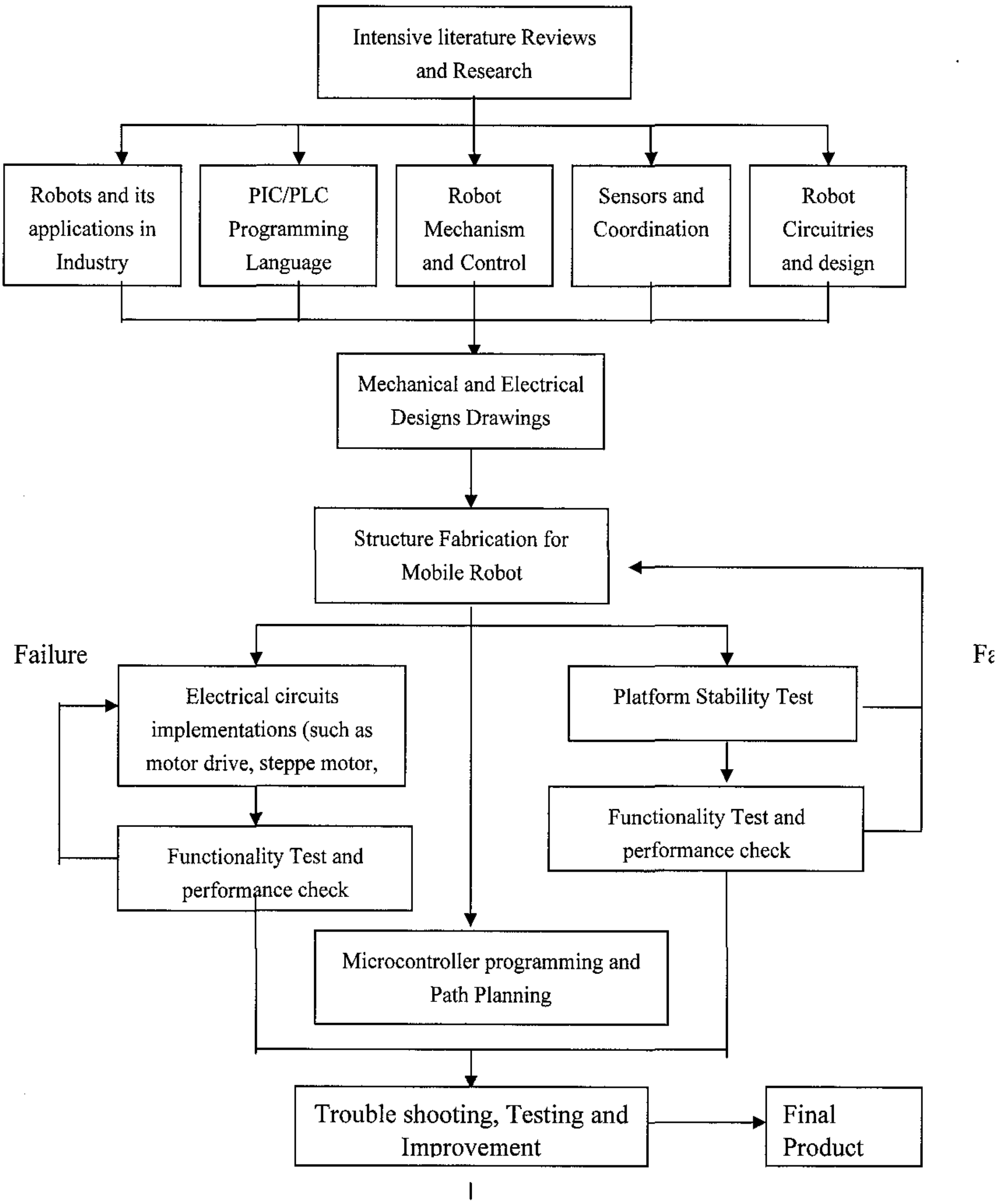


Figure 27 Project Methodology Flow Chart

3.2.1 Literature Reviews:

Intensive literature reviews has been done on mobile robot fabrication. Resources from related books, internet and online journals have been access. The reviews are crucial to identify the method, tools and equipments that are needed for implementation of the robot. The literature review covers five important areas which are:

1. Robots and its application in the industry

   -The technology of robots used in various industries and the level of technologies for each of the robots in its job application.

2. PIC Programming language

   -Learning the programming language (C++) which is used to programme the microcontroller selected for this robot design, which is the Programmable Input Controller (PIC) chip.

3. Robot Mechanism and Control

   -Mechanical Trajectory and dynamics calculation has been done to identify the type of locomotion to be used, the torque of the motor needed and mechanisms that would be implemented on the design

4. Sensors and Coordination

   -Identify the best suited sensor for obstacle avoidance of the robot as well as path planning needed for covering the area of land that needed to be covered.

5. Robot Circuitries and Designs

   - Electrical circuits such as stepper motors and driver circuits for controls of the mobility of the robot.

3.2.2 Mechanical and Electrical Designs

The mechanical structures are drawn out which includes the details in which the real robot would resemble. The electrical designs are done for simulation to test the theory and check the sufficiency of voltage and current supplied to the circuitry.

3.2.3 Structure Fabrication for mobile robot

The wheels are mounted to the robot base. The stepper motor is coupled to the caster wheel which would be used for direction control. Electrical implementations are carried out parallel to the construction of the robot.

3.2.4 Microcontroller Programming and Path Planning

The microcontroller is the 'brain' of the robot. Programming is needed to be done in order to control the movement and path planning for the robot. The microcontroller is implemented on the robot once the robot is completed.

3.2.5 Troubleshooting and Testing

Troubleshooting is needed to ensure a functioning robot is produced at the end of the project period.

3.3 METHODOLOGY OF PROTOTYPE CONSTRUCTION

3.3.1 Chassis Construction

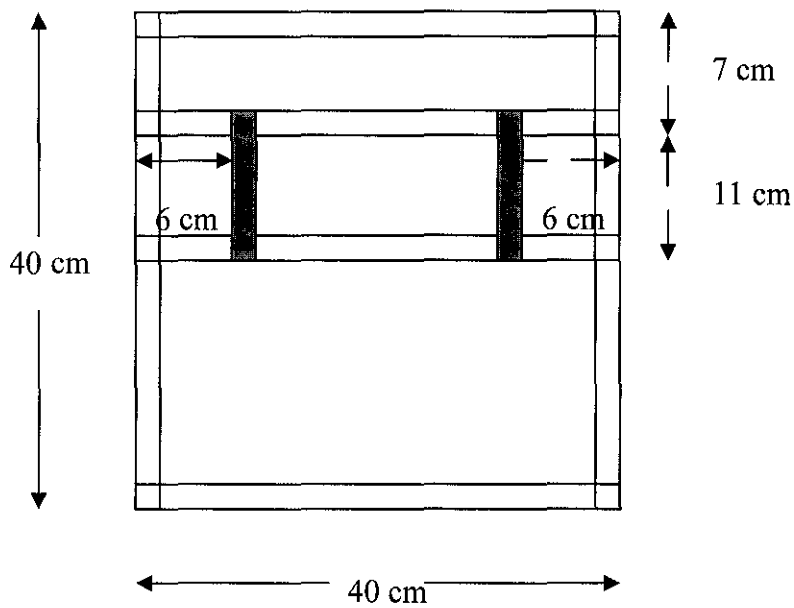


Figure 28 Chassis Structure for mobile robot

The chassis of the robots is constructed from L-shape bars. Due to the scarcity in the amount of L bars in the lab, the L bars used are taken from the scrap metals from the Robocon workshop with permission from the Robocon team and lecturer. The L-shape bars are cut in 40 cm by 40 cm to form square shape. L- Shape bars are used because they are able to form stable and solid structure. The highlighted area in the design is where the gears of the locomotion drives are located.

3.3.2 Mounting the Caster wheel on the Stepper motor

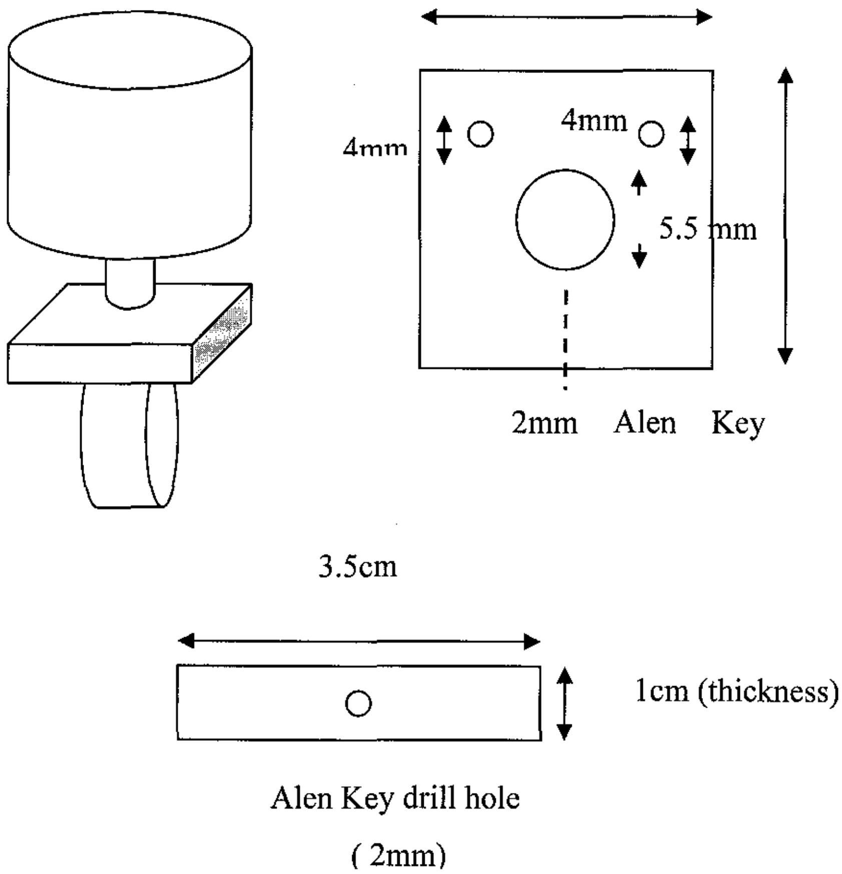


Figure 29 Stepper Motor Coupling

A piece of Aluminium metal of 3.5cm x 3.5 cm x 1cm was cut in the workshop. The piece of aluminium is to couple the stepper motor with the caster wheel. The 4mm drill holes are screwed to the caster wheels while the 5.5 mm drilled hole is to be mounted on the moving rod of the stepper motor. The moving rod of the stepper motor has a diameter of 5.4 mm. Therefore, for best performance, a 2mm Alen Key screw hole is drilled at the side the metal. The Alen Key screw is then used to tighten the connecting between the rod with the coupling when inserted in the 5.5 mm hole.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 THE OVERALL DESIGN

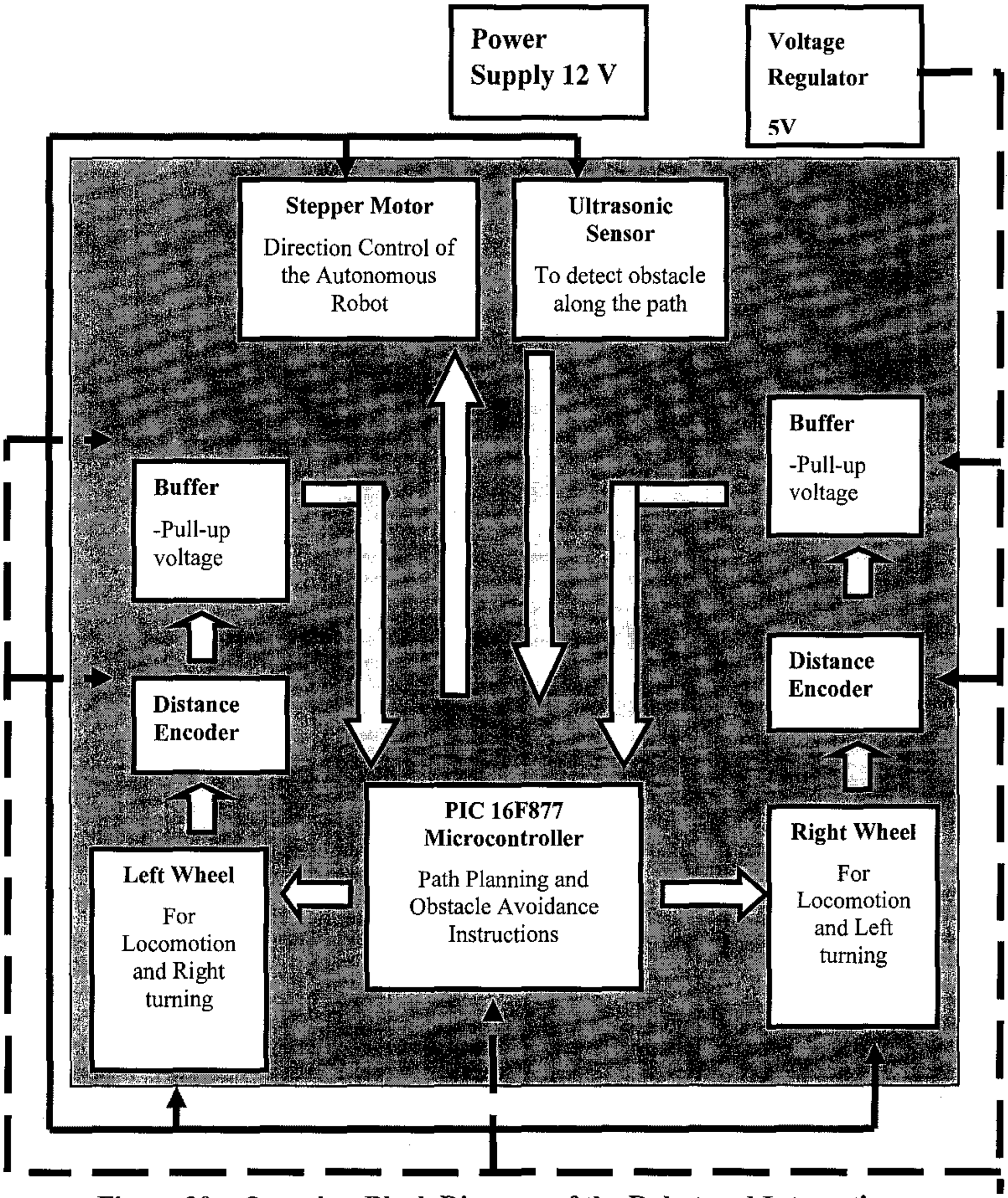


Figure 30 Overview Block Diagram of the Robot and Interactions

With reference to Figure 31, the structure is designed to have one stepper motor at the front for direction control and at the rear is attached with 2 DC motors for locomotion. The chassis of the robot is constructed with aluminum bars for firm and solid foundation for the robot.

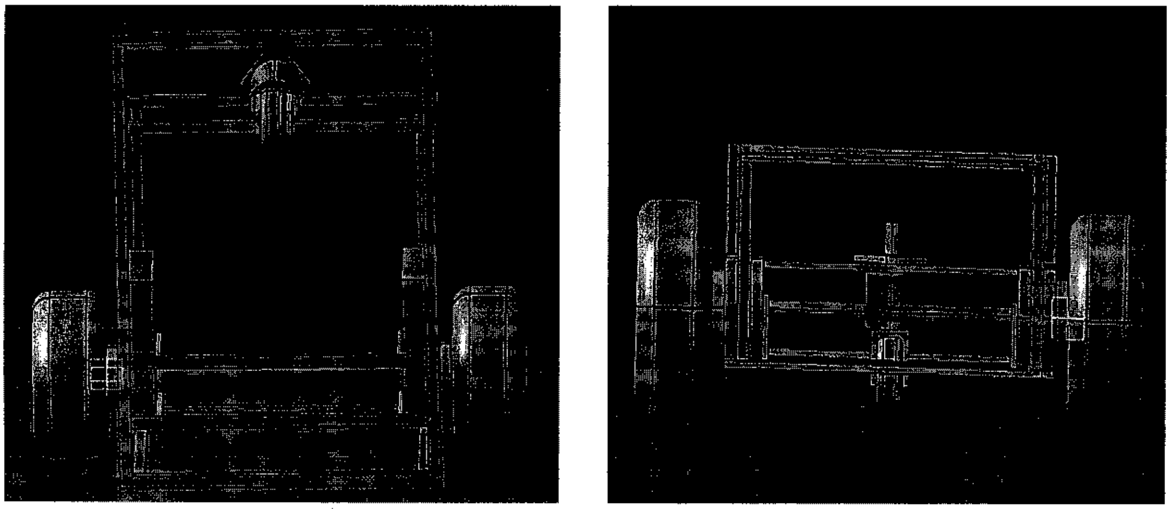


Figure 31 Chassis Structure drawn with Catia ®

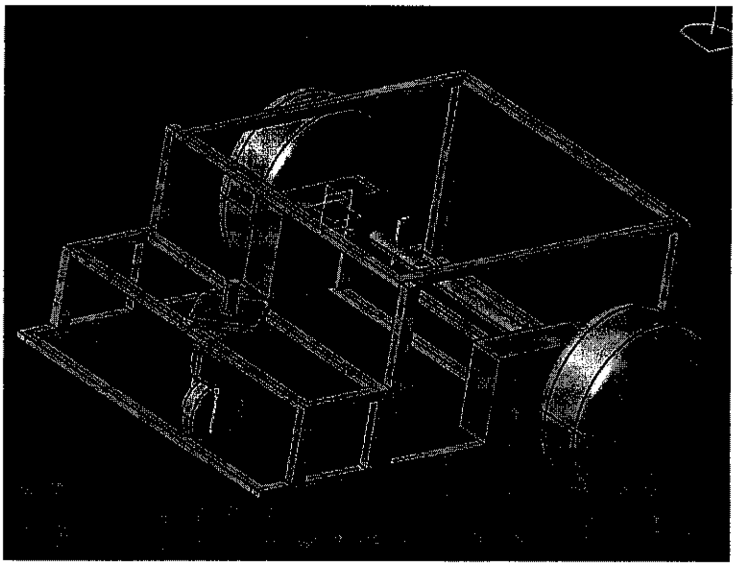


Figure 32 Isometric View of the Mobile Robot

The robot is constructed with two levels, the bottom is used to insert the DC motors and its gearbox and the stepper motor in which directs the robot follow the predetermined path. The blade for the grass cutter is also installed in the lower storey. The upper level is the circuit compartment, in which all the circuitries such as the PIC controller, the distance encoder and the stepper motor translator would be located.

4.1.1 Component Functions and Interaction

With the reference to the Block Diagram shown in Figure 30, the functions of the components are described below:

PIC 16F877 Microcontroller:

The overall process of the robot is determined by this chip. The chip receives inputs from the sensors and processes it to execute appropriate instructions according to the program written by the programmer. The controller also control the drives such as the DC motor according to the path planning codes written

DC Motors

The DC motors are used to move the robot about. The motor has sufficient amount of torque in order to successfully move the whole structure.

Stepper Motor

The stepper is coupled to the caster wheel and used for direction control.

Distance Encoder

The distance encoders used are the Hamatsu Infrared Sensors. The wheel is divided into 60 sections, in which 1 section would cover 1cm of traveled distance. Therefore, in order for the robot to move one meter of traveled distance, the encoder must detect 100 pulses and inputted to the microcontroller. The output HIGH for the Hamatsu sensors is 1.2 V which is not sufficient as input for the microcontroller. Therefore, a pull-up circuit is needed to output a 4.6 V.

Ultrasonic Receiver / Transmitter

The ultrasonic sensor is essential for the obstacle avoidance feature in this robot. When an obstacle is detected along the predetermined path of the robot, a signal would be sent to the microcontroller by the ultrasonic sensors. The obstacle avoidance program would then be executed to avoid any collision of the robot.

Stepper Motor Translator

Stepper motors require certain combination of pulsing in order to move. The stepper is able to rotate to a certain angle which is suitable for direction control.

4.1.2 Pictures of the Constructed Robot

Figure 33 Front view of the Robotic Lawn Mower

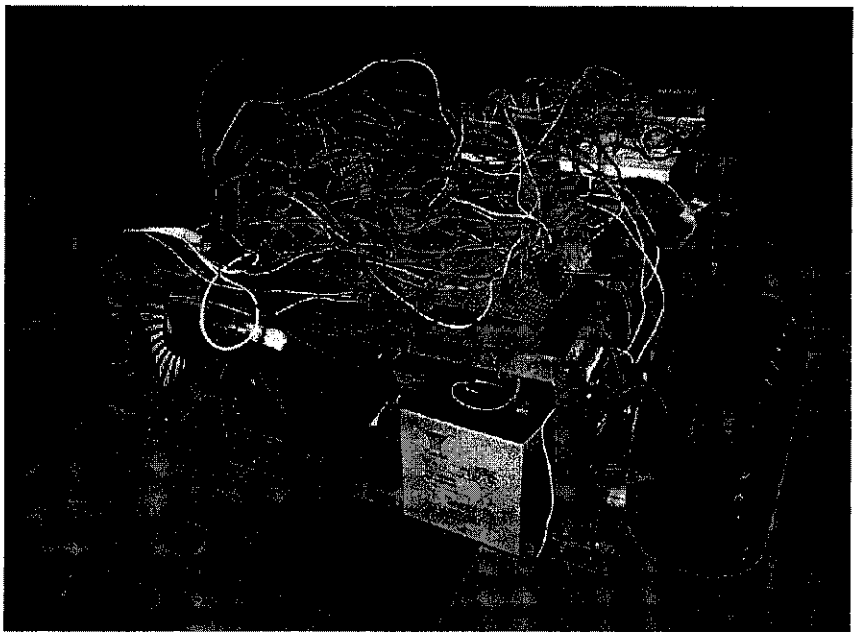


Figure 34 Rear view of the Robotic Lawn Mower

4.2 ROBOT PATH PLANNING AND MOVEMENT

The project incorporates motor locomotion control, direction control, blade cutter and collision avoidance. The PIC 16F877 has 5 Ports in which can be defined as either inputs or outputs. There are sufficient outputs to drive the whole robot without needing extra microcontroller to be slave controllers. With reference to Figure 33, the implementation of smaller microcontrollers as slave controllers would definitely reduce the complexity of the program. However, the dependency between the master and slave controllers would reduce the reliability of the system in the robot.

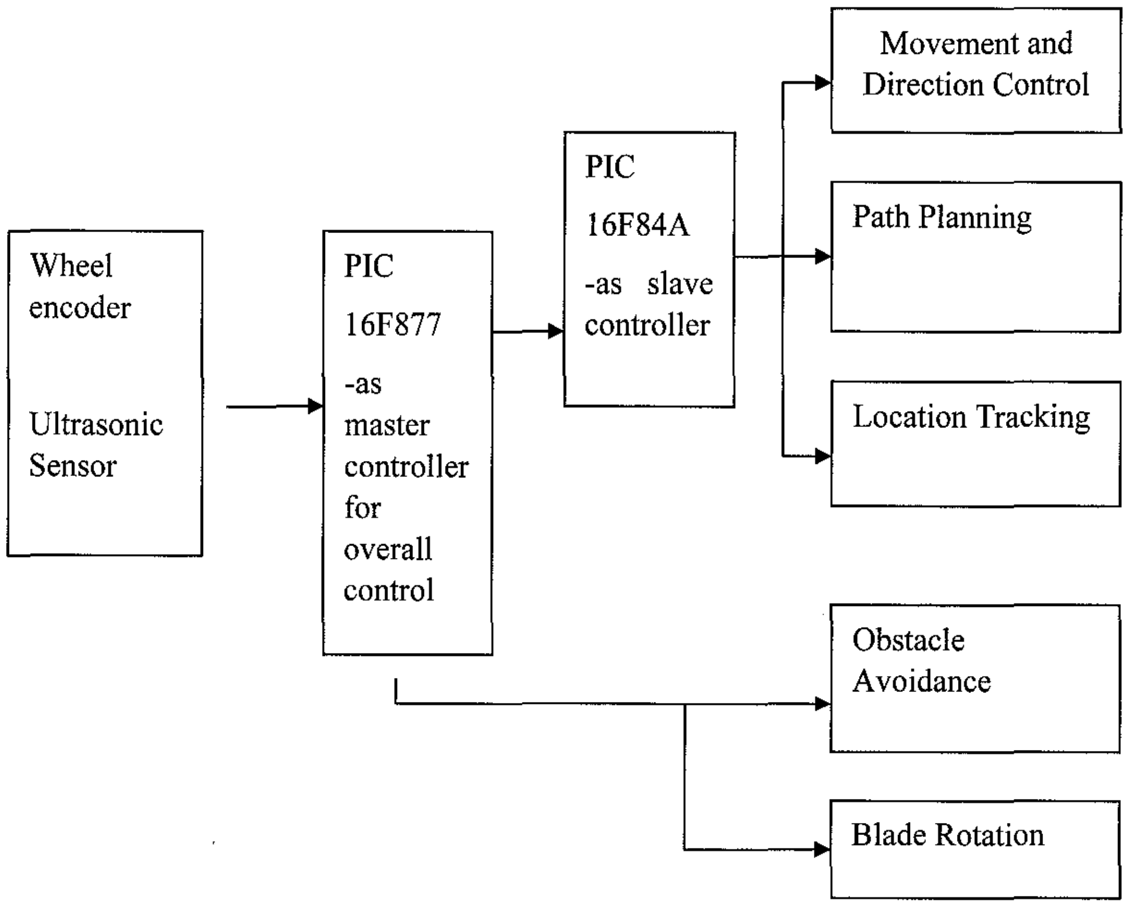


Figure 35 Microcontroller Interaction Block Diagram with Slave controller

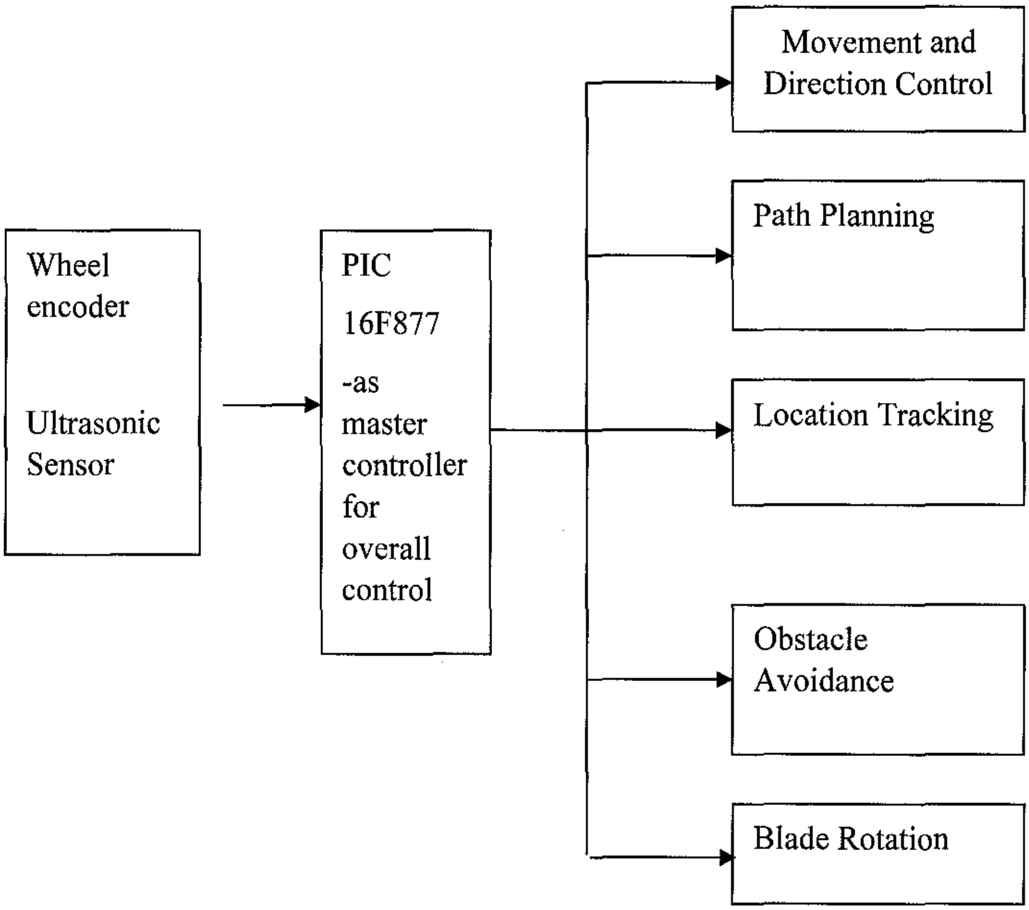


Figure 36 Microcontroller Interaction Block Diagram without Slave Controller

Therefore, the Microcontroller Connections without the usage of slave controllers was selected for this project as there were sufficient pins for inputs and outputs and also reduce cost. The 16F877 also has PWM pins which can output Pulse width Modulation pulses for motor speed control.

4.2.1 Block Functions Operation

Direction Control and Locomotion

Both the stepper motor and DC motor must collaborate correctly in order to move straight, reverse and perform an accurate turn to the left or to the right.

Path Planning

This allows the robot to follow a predetermined path set by the user in order to move within the barrier defined and to cover the whole area of grass to be mowed.

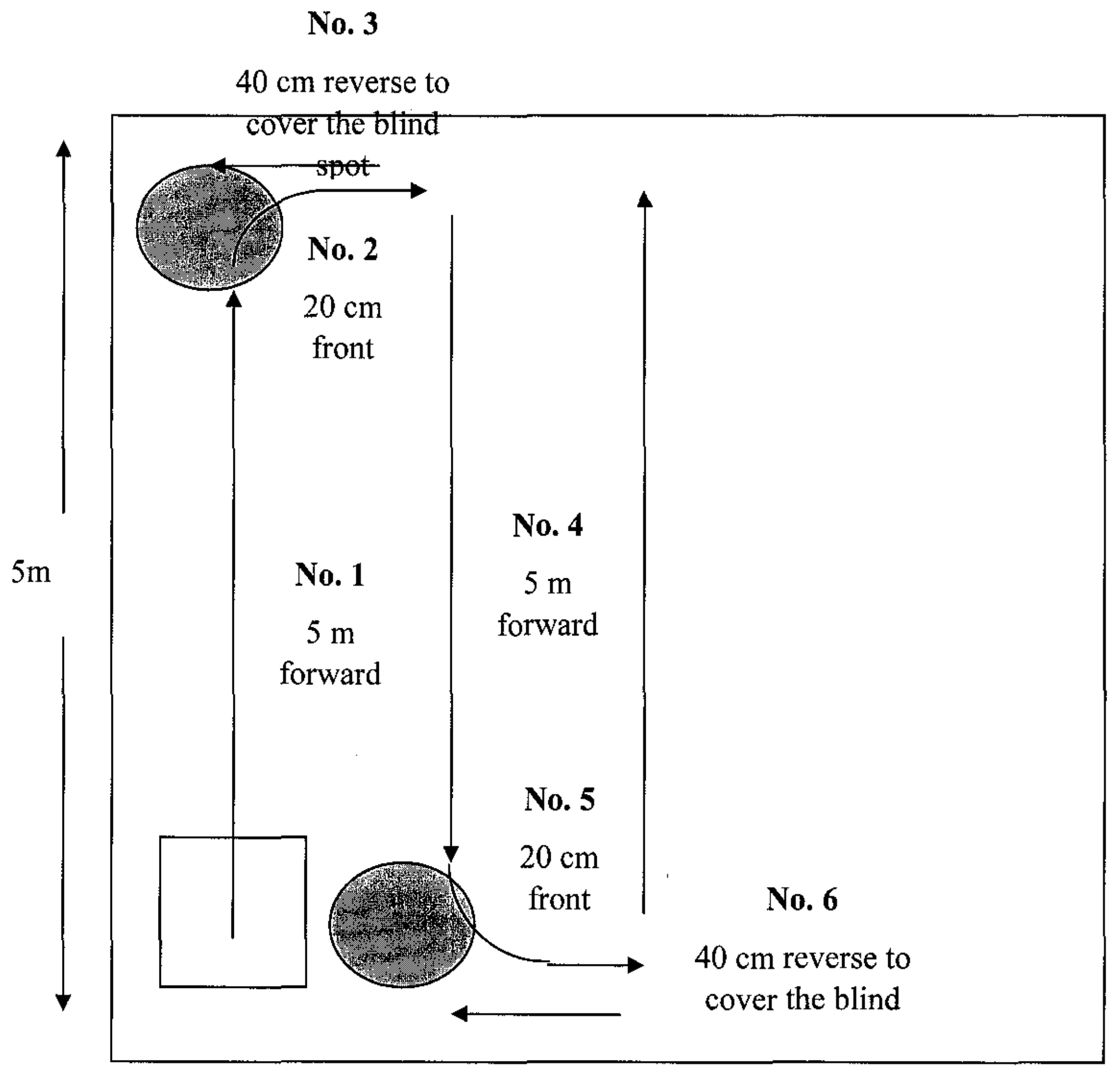
Location Tracking

This programme block is essential to inform the robot of its current location. For example, if the robot is programmed to move 5 m forward, the location tracking program is to calculate whether the robot has already arrived at the destination. This is done through the counting of pulses using the distance encoder.

Obstacle Avoidance

This programme block is needed to avoid any obstacle within the path. The robot should change direction and move around the obstacle , arriving back to the original predetermined path and continue its journey.

4.2.2 Predermined path and Location tracking



The predetermined movement path is described below:

- The robot starts at the left side of the field and moves straight for 5 meters, cutting the grass along the path. (No.1)
- When the robot distance encoder calculates that the robot has already traveled 5 meters, the robot would stop and turn right and cover 20 cm front. (No.2)
- However, when the robot turns, there is a blind spot marked by the gray circle, therefore, the robot must reverse by 40 cm in order to mowe the grass in the greyscale area. (No.3)
- After doing so, the robot would then turn right again and cover the next 5 meters. (No.4)
- The robot would finally reach the end of the 5 meters and turn left. Moving

forward 20cm (No.5) and reversing again to cover the blindspot area (No.6).

- The robot finally turns left, returning to the similar position of the starting point of the robot. Thus the whole predetermined path cycle repeats.

4.2.3 Obstacle Avoidance and Distance Correction

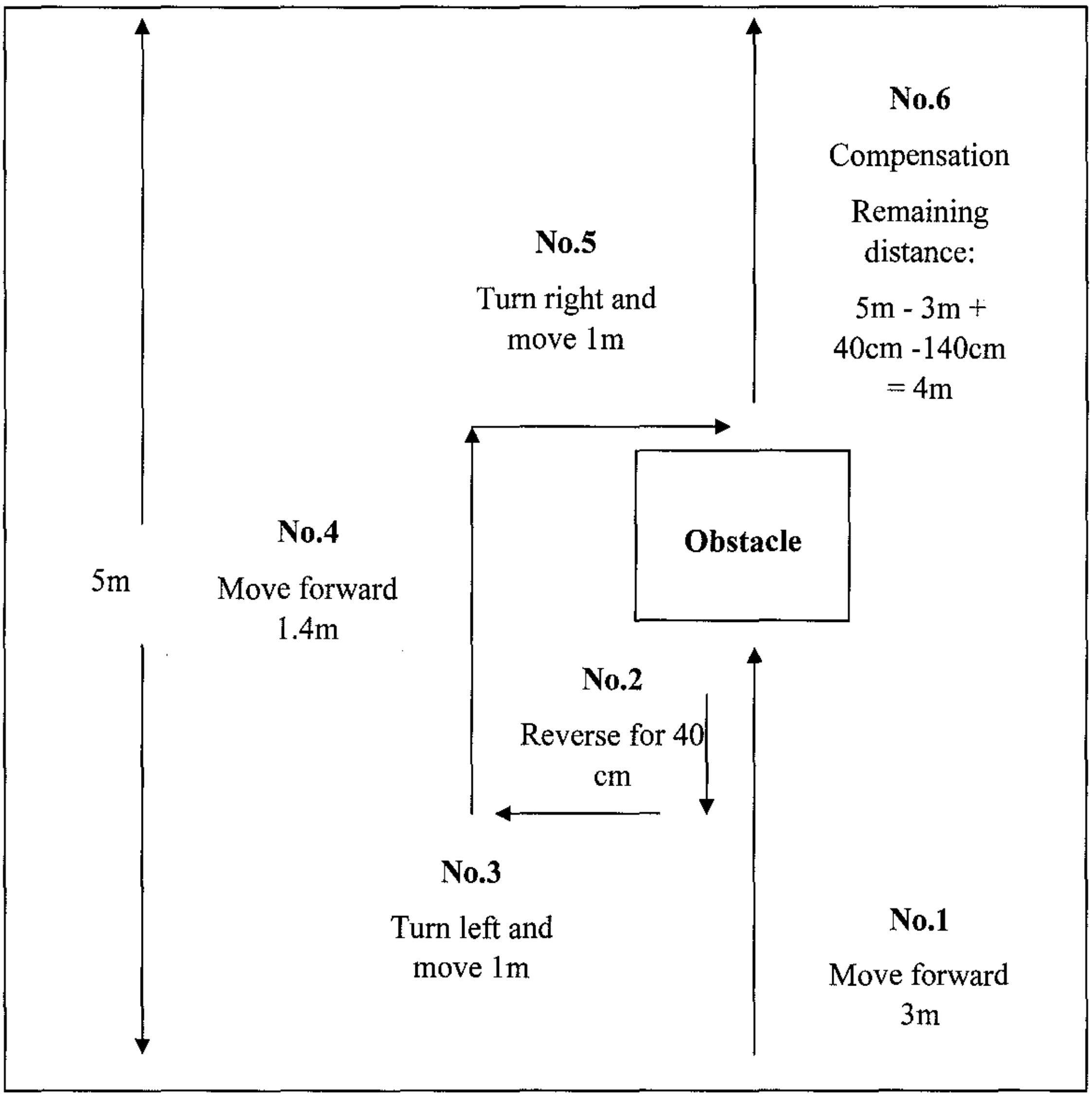


Figure 39 Obstacle Avoidance Plan

- For the obstacle avoidance system, assuming that the robot is path planned to move straight for 5 meters. An obstacle is placed within its path (3m). Referring to (No.1)

- When the robot detects the presence of an obstacle via the ultrasonic sensor, the robot would reverse for 40 cm.(No.2)
- The robot would then turn left and move for 1 m. (No.3)
- After doing so, the robot would then turn right and move 1m to go around the obstacle plus an additional 40 cm which is to compensate the 40 cm of during the initial detection. (No.4)
- The robot would turn right and return to the original predetermined route. (No.5)
- However, compensation algorithm must be included, instead of the remaining 2 m that the robot has to cover, the robot must subtract 1m for the collision avoidance system. Therefore the remaining distance to travel is 1m.

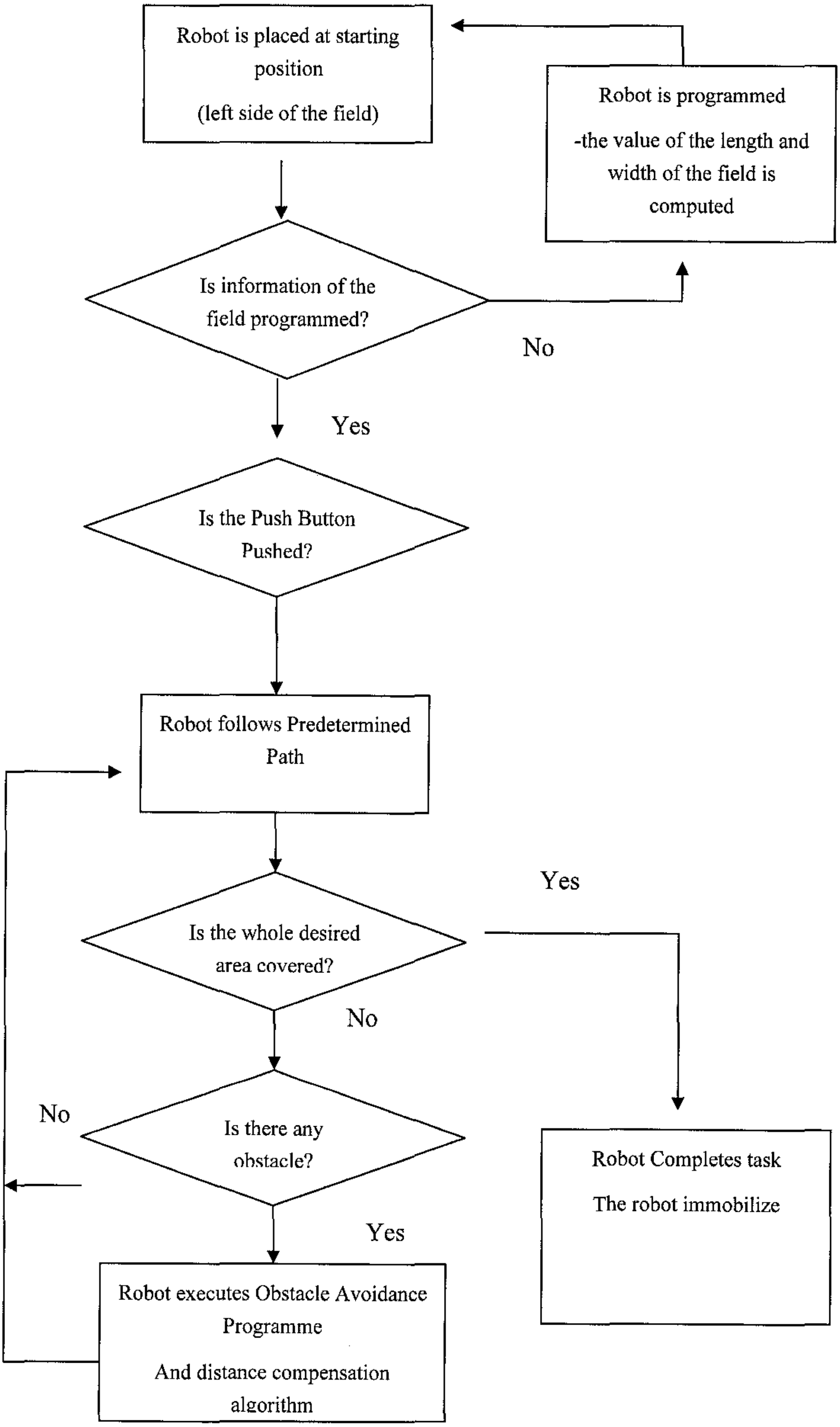


Figure 40 Flowchart of Autonomous Mobile Lawn Mower Robot with Obstacle avoidance System

4.3 CHASSIS CONSTRUCTION

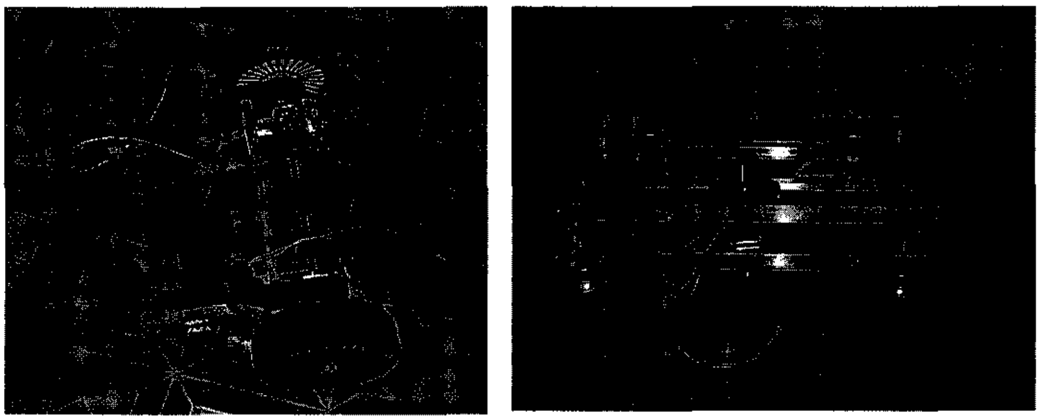


Figure 41 Chassis Structure (dismantled due to instability)

The above chassis was constructed initially. The robot is made using the L-shape bars. The chassis was constructed at 33cm x 33cm. however, when the gear ratios were fitted onto the platform, the platform was slightly too small. There were issues with stability as most of the connections were joined by screws and nuts. The screws would come loose after a period of time. The above structure was dismantled and a new platform was constructed.

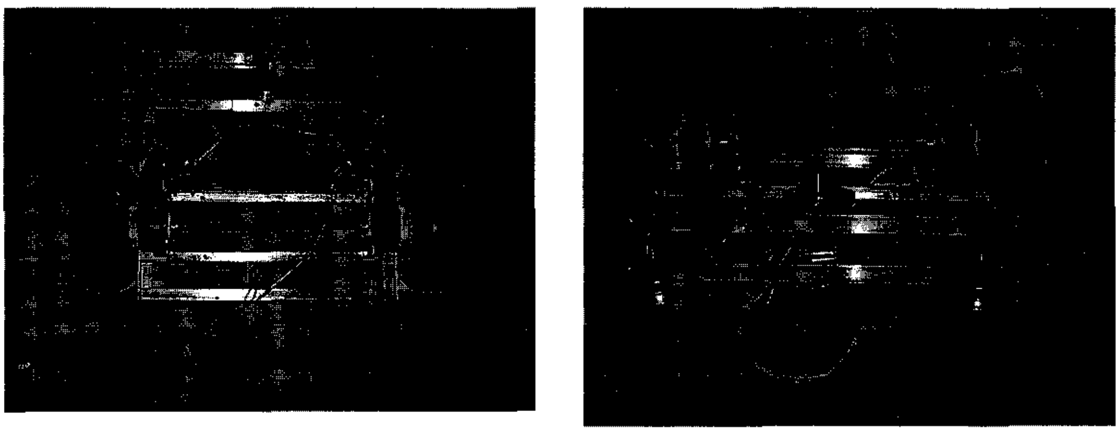


Figure 42 Improved Chassis Stucture (40cm by 40cm)

Therefore a new platform is constructed at 40cm x 40cm. The joints are connected and hold firm using rivet. The stepper motor is mounted on the front of the robot for direction control.

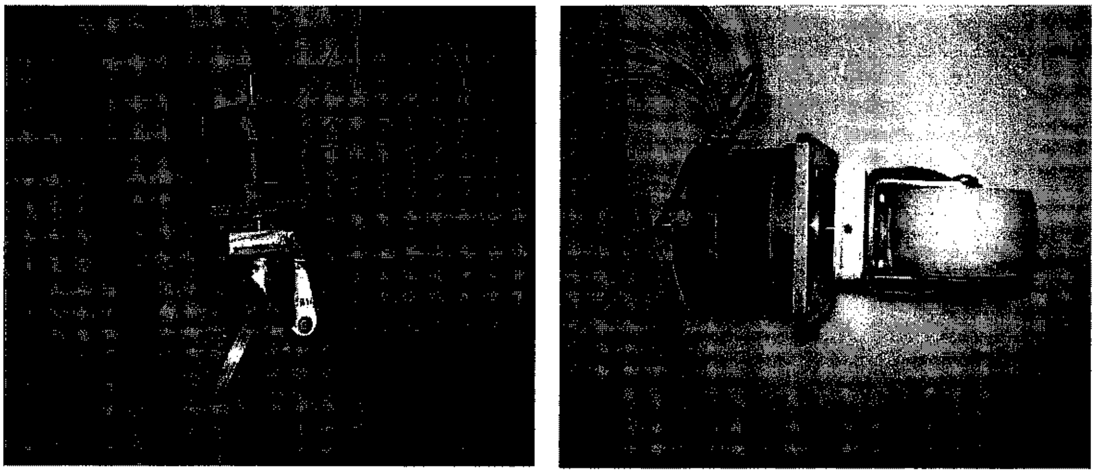


Figure 43 Coupled Caster wheels on the Stepper Motor

The 12 Volt RS stepper motor is coupled on the caster wheel. Screws and nuts are used to tighten to a piece of aluminium.

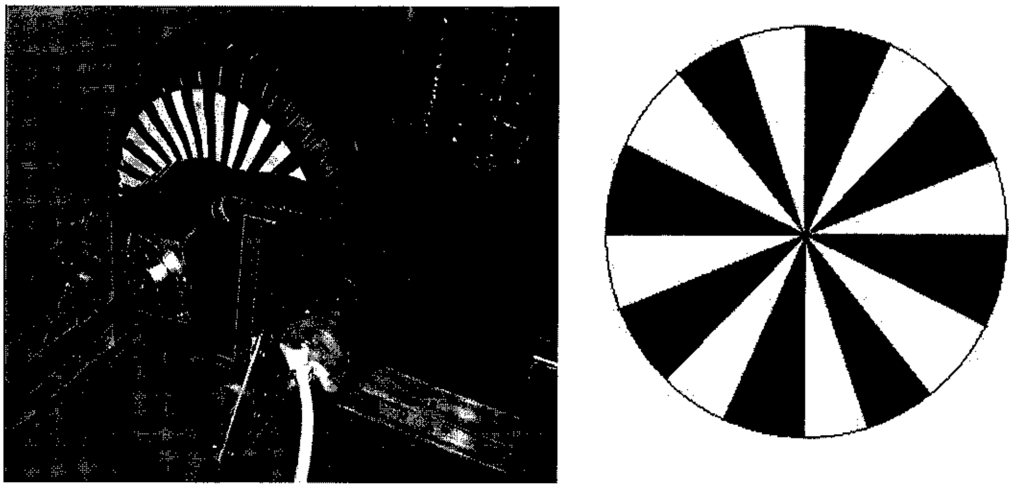
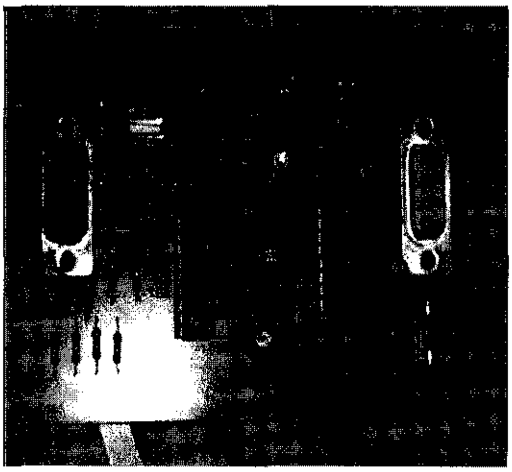


Figure 44 Distance Encoder Disk

The wheels and gear ratio was reused and taken from the ROBOCON lab. The wheel includes a shaft encoder which would be implemented with a infrared photoelectric sensor for distance calculation. The disk is divided in equal slices of alternating black and white. The disk is mounted on the wheel and the photo reflector is placed where the wheel is directly facing the encoder. The diagram above displays a sample of a encoder disk with 16 sections.

4.4 ELECTRICAL COMPONENTS

4.4.1 The PIC 16F84A programming board



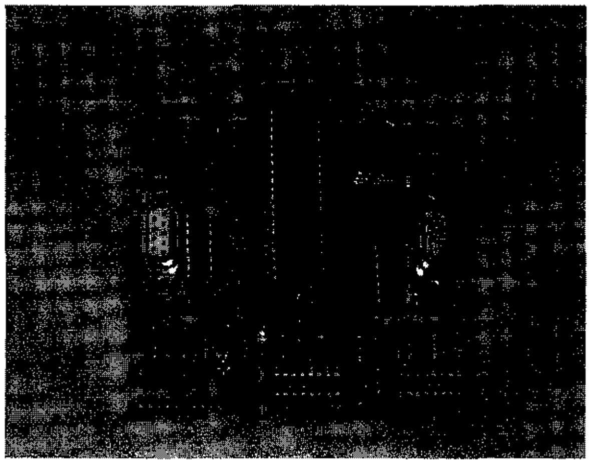


Figure 45 PIC 16F877 Programmer Board and Target Board

The programming board was constructed in order to be able to interface between PIC 16F84A with the microcontroller programmer provided by lecturer Mr. Zuki. The Bumble Bee and the PIC compiler was installed on the computer and programme written is transferred to the chip through the help of the Serial Cables.

4.4.2 The application board for PIC 16F84A

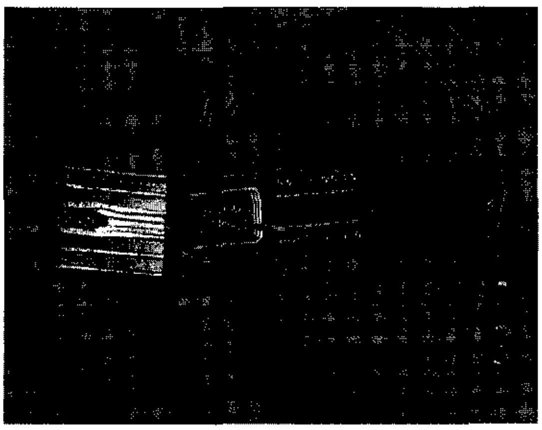


Figure 46 PIC 16F84 Programmer Board and Application Board

The application board for the PIC 16F84A has been constructed and interface with the stepper motor circuit. The application board is then able to interface the PIC chip with other drives such as the stepper motor, the H-bridge controller for the locomotion drive and the sensors for obstacle avoidance.

4.4.3 Application board for PIC 16F877

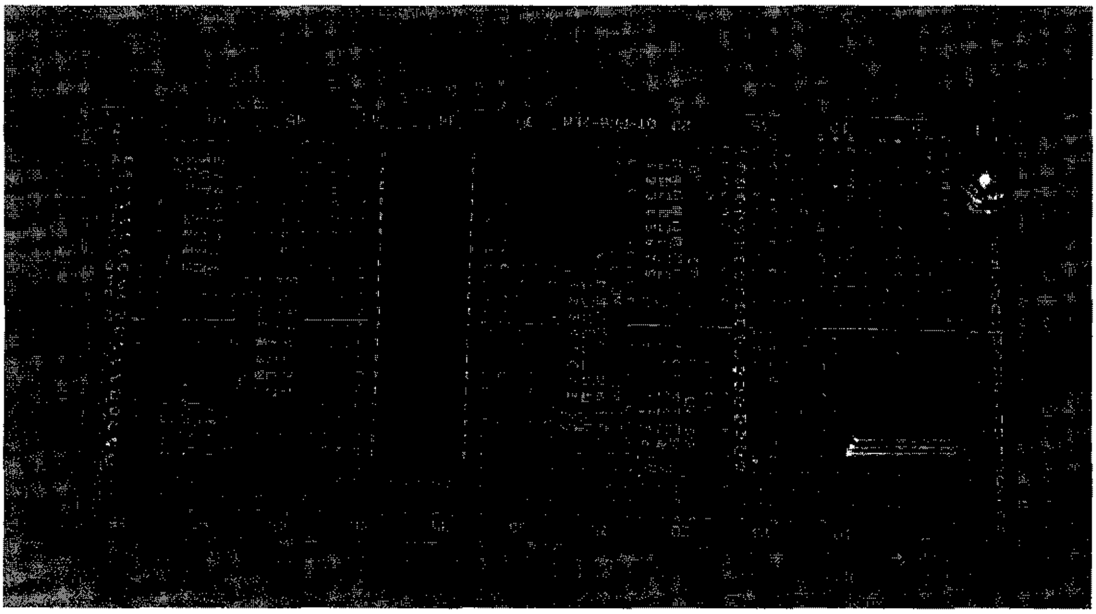


Figure 47 PIC 16F877 Application board with input/output connectors and Switches

The PIC877 is used as the Master interface with the other components for the robot. This is because the PIC877 can store larger memory programmes and it includes Pulse Width Modulation option which allows controlling the speed of the motor. The PIC 877 has more Ports which can be configured as either outputs or inputs.

4.5 Ultrasonic Sensors for Collision Avoidance System

4.5.1 Ultrasonic Receiver and Switching Circuit

The circuit has been designed based on the schematics shared on the internet by the electronic hobby kit webpage. The circuit shown in Figure 46 shows that if the receiver detects ultrasonic waves, the relay switch would activate.

The circuit works based on the ultrasonic transducer when sensing ultrasonic signals. It converts it to electrical input with the same frequency. The purposes of transistors T3 and T4 are to amplify the signal (as they are supplied by 9V voltage source). The amplified signals are then rectified and filtered. The filtered DC voltage is fed to the inverting input of the op-amp IC2. The non- inverting pin is connected to a variable DC voltage in which the threshold value of ultrasonic signal received can be manipulated. The output of op-amp is used to bias transistor T5 which will then bias the transistor T6. T6 can then activate the relay to control other equipment or connected to the microcontroller.

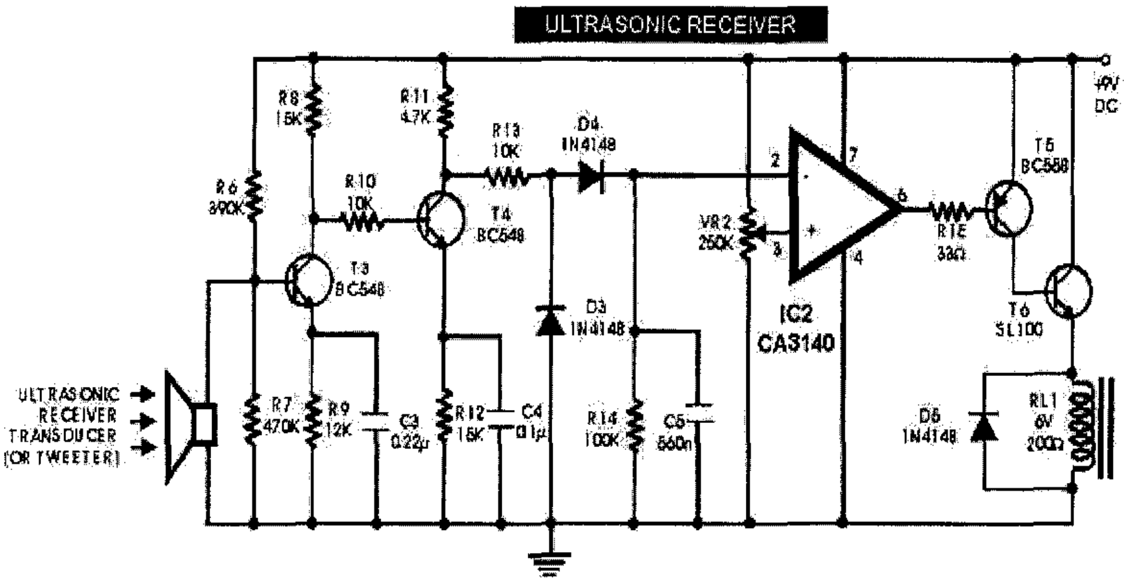


Figure 48 Schematic of Ultrasonic Reciever

The circuit was tested through simulation on the Electronic Workbench ® software and the simulated result was compared to the final product when built. The simulation results are vital for trouble-shooting the actual circuit.

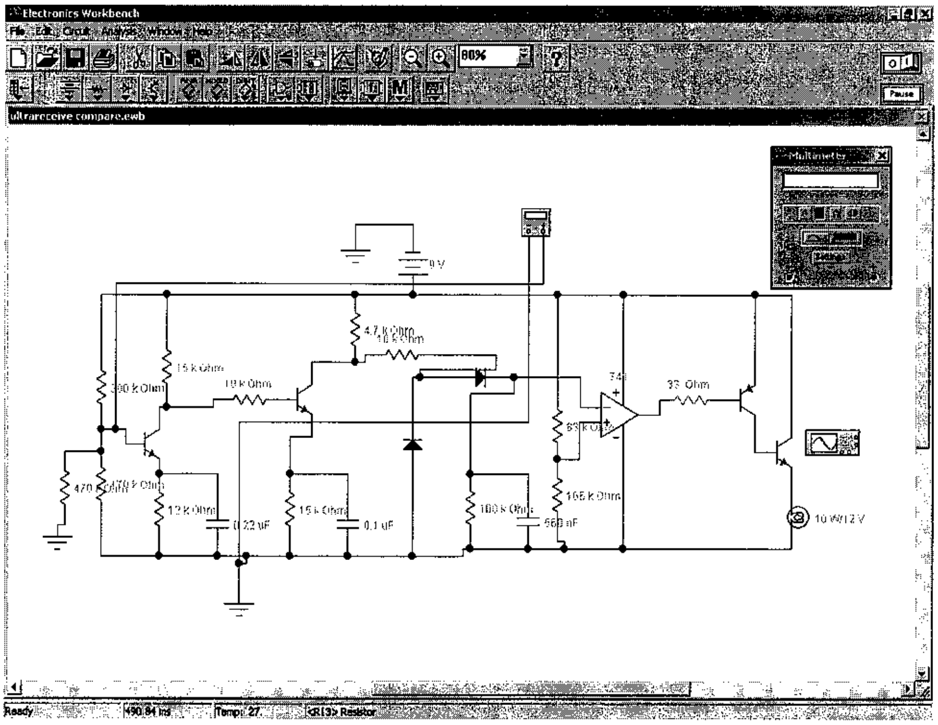


Figure 49 Simulation for Ultrasonic Receiver using Electronic Workbench ®

4.5.2 Ultrasonic Transmitter

The ultrasonic transmitter is constructed using 555 based astable multivibrator. It oscillates at a frequency of 40-50 kHz. The transmitter is powered from a 9-volt or a 12-volt supply. The value of 4.7K with plus/minus of 10K (pin7) and the value of 18K (pin 6) and C1 of 680pico Farad was chosen to generate a 40 kHz to 50 kHz frequency. The accuracy of the frequency is important to be detected efficiently by the receiver. The 40kHz frequency can be calculated using the formula

$$f = \frac{1.44}{(2R_2+R_1)*C_1}$$

$$f = \frac{1.44}{(2*18k+15k)*680pF}$$

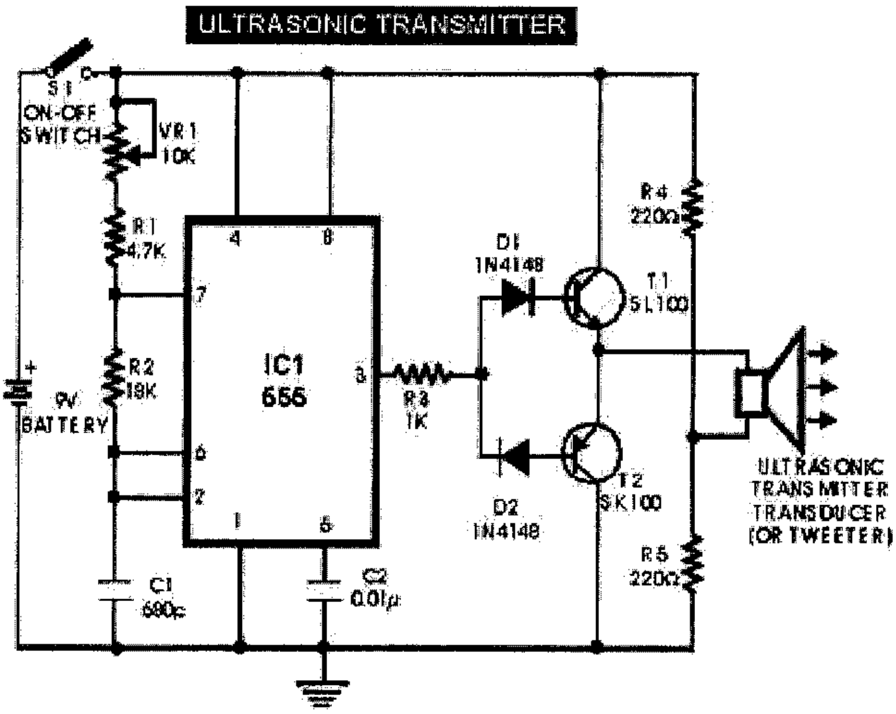


Figure 50 Schematic for Ultrasonic Transmitter using NE555 Timer

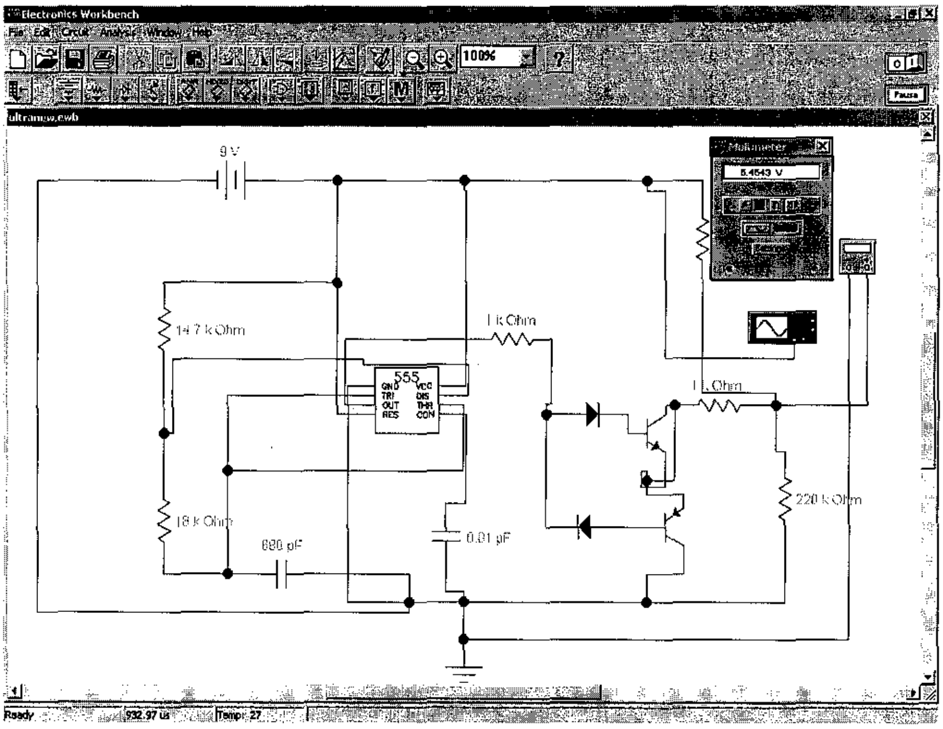


Figure 51 Simulation for Ultrasonic Transmitter using Electronic Workbench ®

The circuit was also simulated in Electronic Workbench ® studio and the results were compared to the actual design. The output signal was generated at 40 kHz as shown from the output signal from the oscilloscope of the simulation tool.

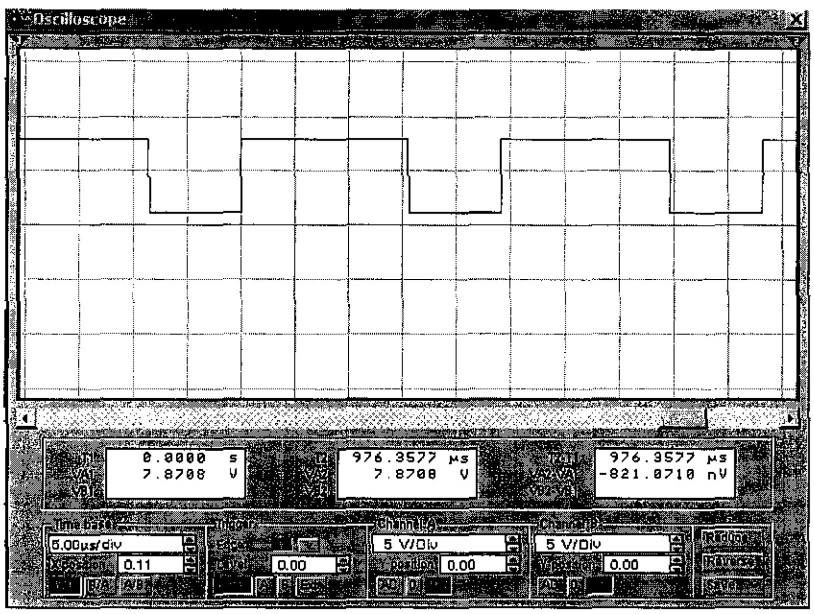


Figure 52 Simulation for 40 kHz frequency generation

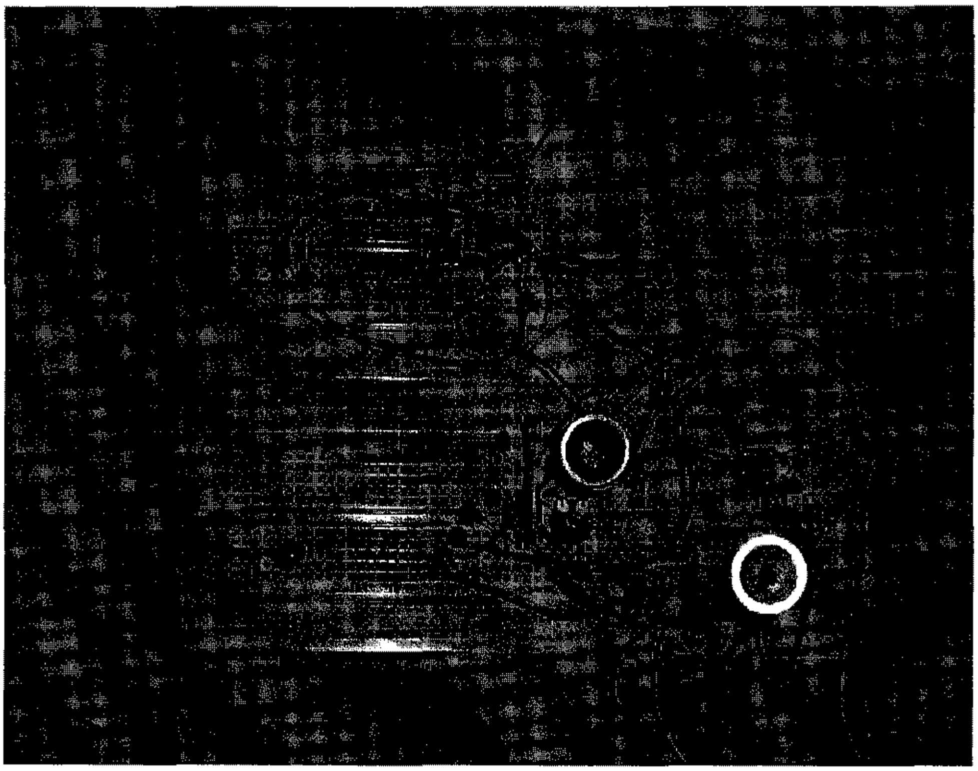


Figure 53 Actual Circuitry for Ultrasonic Sensors.

4.6 Locomotion Drive

4.6.1 Stepper Motor control Circuit

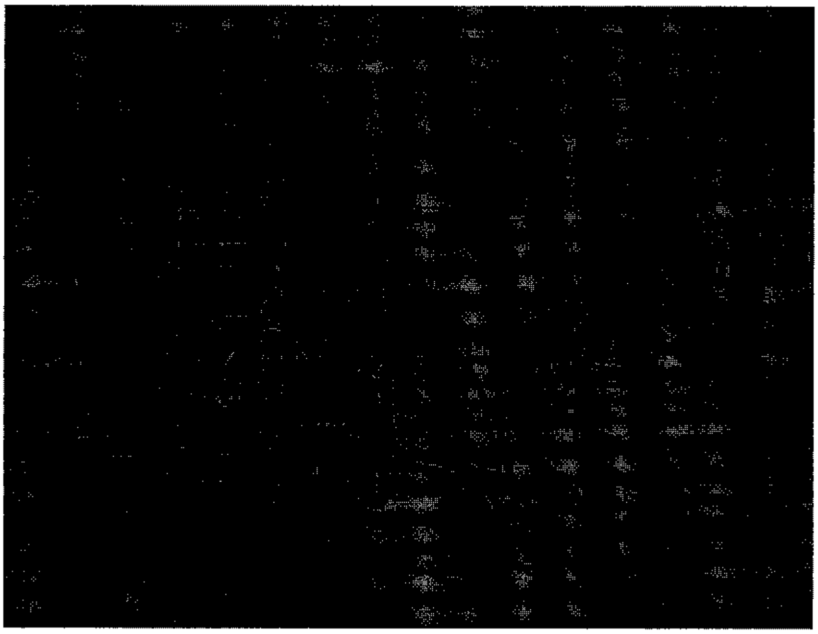


Figure 54 Stepper Motor Control circuit using J-K Flip Flop

The s The stepper motor control circuit has been built using JK flip flops and XOR gates. The circuit is able to operate the stepper motor in the forward and reverse direction. The design was taken from the McComb, Gordon. Robot Builder's Bonanaza: 99 Inexpensive Robotic Projects, USA, McGraw-Hill. The circuit was simulated using the Electronic Workbench software beforehand to understand how the circuit functions. However, the stepper motor is too small and does not provide enough torque to turn the caster wheel. Therefore the 12V RS stepper motor has been selected and implemented on the mobile robot.

4.6.2 Using the EDE 12400 bi-polar stepper motor controller

The previous stepper motor controller produced using JK flip-flop is quite unstable. Therefore the EDE 12400 bi polar stepper motor is used. The reason for the use of this module is that is has higher tolerance for larger current that may pass through to the motor. The specifications are as shown below:

- Max. motor voltage 46V
- Max. current 2 Amps per coil
- Full/half stepping and direction control
- Complete stepper motor control unit
- Based on the proven L297/L298™ chipset

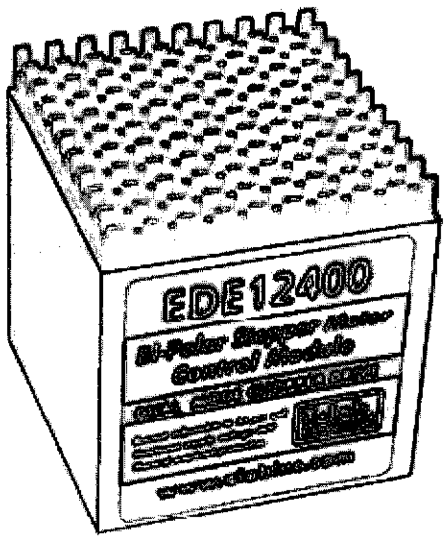


Figure 55 Diagram of the EDE 12400 module

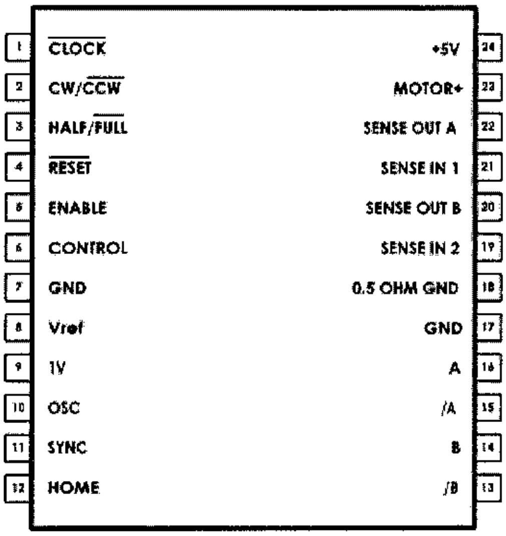


Figure 56 Pin layout of the module. The details are listed in the datasheet

Similar to the stepper motor controller, the module would pulse the motor with the output from pin 13 to pin 16. To operate it in a certain direction, the stepper has to be pulsed in a certain pattern. Pulsing in a opposite direction combination will make the stepper to go in the reverse direction.

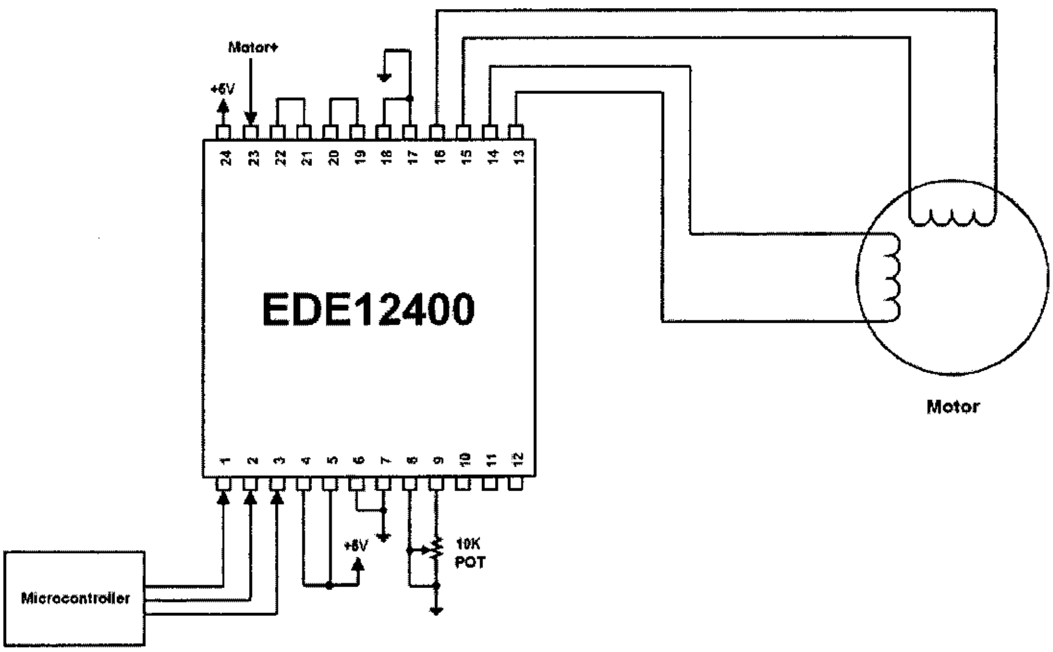


Figure 57 Connection layouts for the EDE 12400

The Stepper Motor is connected to the EDE 12400 module. The stepper motor rotates when receives input from the microcontroller. Since the Stepper Motor has a locking system, the angle of the wheel would not change unless a pulse is received from the controller. Thus the robot is able to move according to the direction of the caster wheel.

The EDE12400 is able to control stepper motor at coil currents up to 2 Amperes. When connected as shown in Figure 3, the EDE12400 module will operate the motor based upon the inputs of the CLOCK, CW/CCW, HALF/FULL, and RESET pins.

RESET mode (pulsing of stepper)

When RESET is set to high, a low-going (+5V to 0V) pulse on the CLOCK input will cause the motor to rotate one step at the low-to-high transition of the pulse. Thus turning the reset high and low would pulse the stepper motor.

Clockwise and Anticlockwise movement of stepper

The CW/CCW (clockwise and counter-clockwise pin) determines the direction of shaft rotation.

Half Mode and Full Mode

The HALF/FULL pin determines whether to use full-step drive sequence (providing a 1.8°/per step rotation) or a half-step drive sequence (providing a 0.9°/per step rotation).

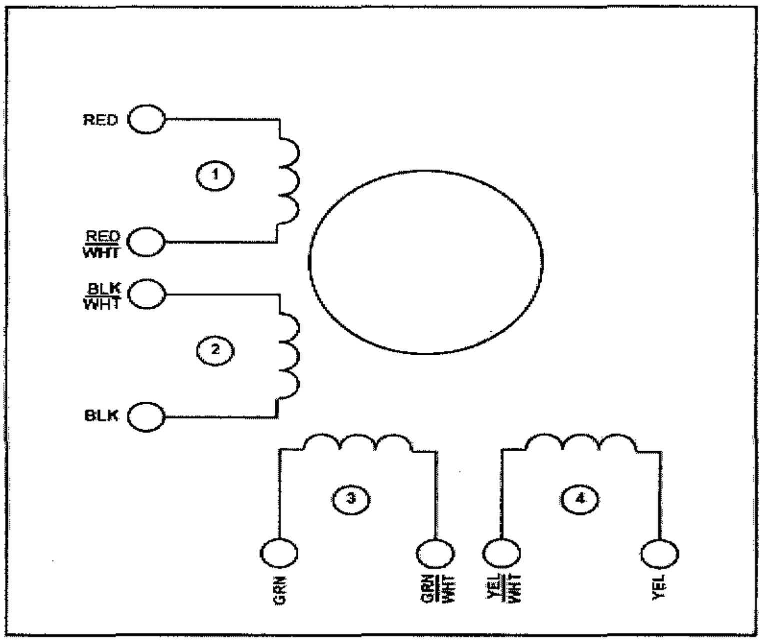
Driving Output to the Stepper Motor

The output drive (A, /A, B, /B) connects to the input of the stepper motor

In which the stepper rotate.

4.6.3 Stepper Motor Wire Connection

8 Wire configuration



Exciting sequence and direction of rotation when facing mounting flange end.						
Step	Red	Green	Black	Yellow	Com	CW
1	On	On				
2		On	On			
3			On	On	+dcV	
4	On			On		

Figure 58 Wire Connection to the 12 V 1.8 degree bi-polar Stepper Motor

The diagram above shows how the stepper motor should be connected. Connecting the wires wrongly would create short circuit and can permanently damage the motor and the h-bridge circuit.

Table 2 Pulsing Sequence for Stepper motor Direction Control

START POINT →

SEQUENCE REPEATING →

Step Number	Q1	Q2	Q3	Q4
	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
5	ON	OFF	ON	OFF

COUNTER CLOCKWISE ↑

CLOCKWISE ↓

The Stepper motor has to be pulse at a certain sequence for proper direction control. Every successful pulse would move the motor by 1.8 degree either clockwise or anti-clockwise depending on the pattern on which its is pulsed. For example, if the stepper motor receive a pulse on Q1 and Q4, followed by Q1 and Q3, Q2 and 3, the stepper would rotate clockwise with 1.8 degree change for each pulse.

However, if the stepper starts with a pulse on Q1 and Q4, followed by Q2 and Q4, Q2 and Q3, the stepper motor would move anti-clockwise.

The diagram below shows the EDE 12400 soldered to a veraboard for application purpose.

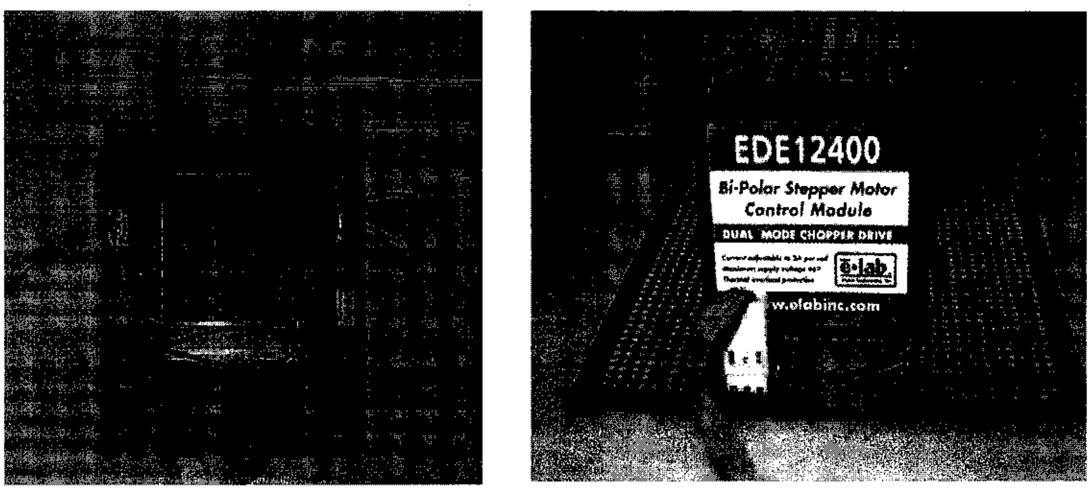


Figure 59 The EDE 12440 soldered on the Veraboard and ready for application

4.6.4 *The H Bridge Circuit*

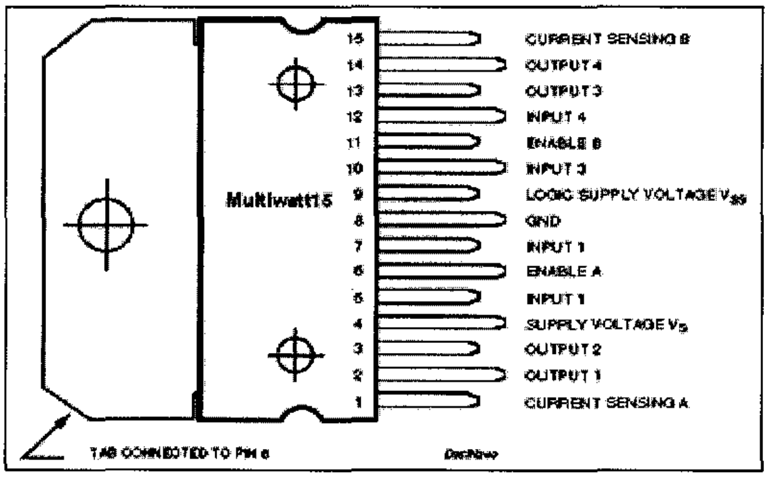


Figure 60 Pin specification of the L298N H-bridge controller chip

The L298N consist is an integrated monolithic circuit which is a high voltage, high current dual full-bridge designed to accept TTL logic levels and drive inductive loads. The L298 has the following specifications:

- Operating supply of up to 46V
- Total DC current up to 4A
- Over temperature protection
- Logical input up to 1.5V

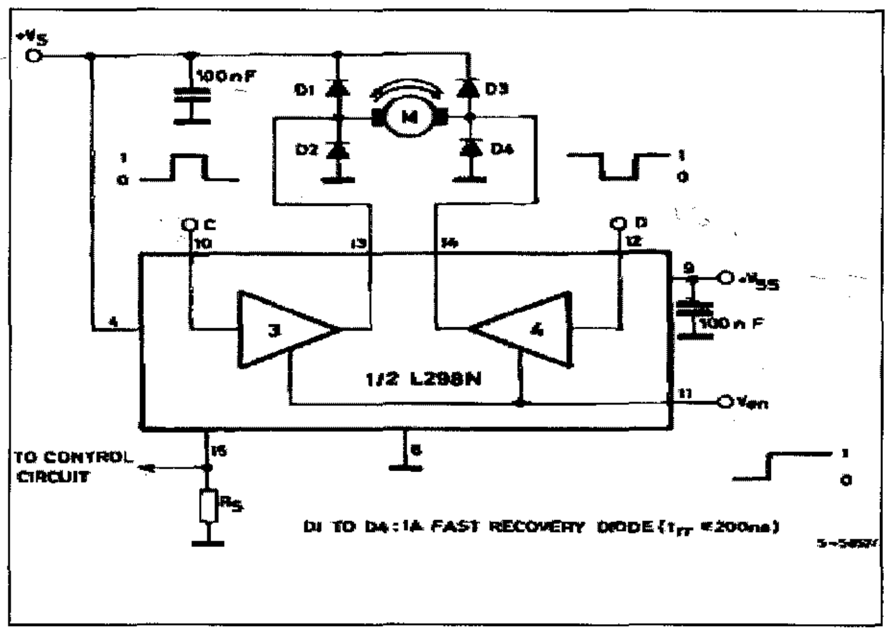


Figure 61 Example of the connections for a H-bridge motor drive circuit

The schematic of the motor circuit in Figure 5 shows how the L298N controls the drive train of the motor. There are 6 bits used to control two motors. Pin 10 and Pin 12 determines the direction of the motor, whether to move in the forward or reverse direction. Pin 11 however, would determine if the motors are on or off. Thee speed of the motor will be controlled by PWM from the microcontroller since the duty cycle will determine the percentage of the motor being turn on during one cycle.

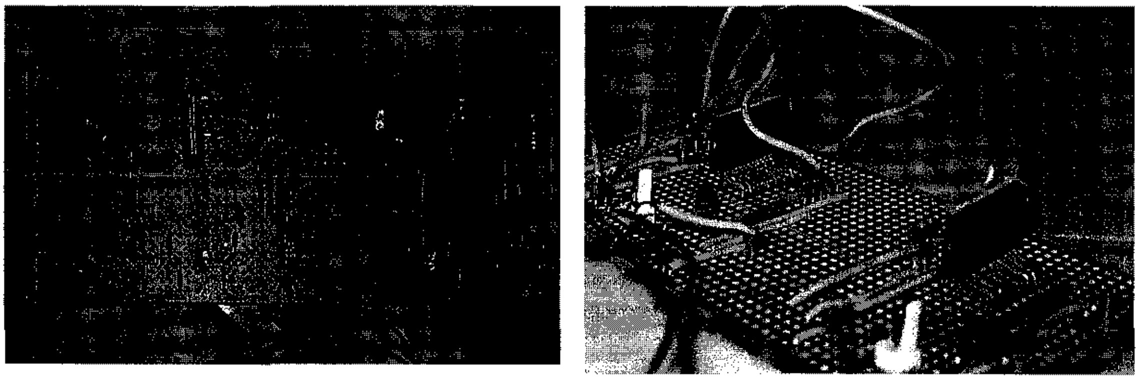


Figure 62 The L298 H-Bridge soldered on the Veraboard and ready for application

Table 3 H-Bridge operation true table

Motor rotation (mode)	Pin 10	Pin 12	Pin 11 (on/off)
Clockwise (forward) (maximum speed)	High	Low	100% duty cycle
Anticlockwise (reverse) (maximum speed)	Low	High	100% duty cycle
Hard Stop	High	High	100% duty cycle
Free wheeling Stop	Low	Low	100% duty cycle
Clockwise (forward) (half of maximum speed)	High	Low	50% duty cycle
Anticlockwise (reverse) (maximum speed)	Low	High	50% duty cycle
Hard Stop	High	High	50% duty cycle
Free wheeling Stop	Low	Low	50% duty cycle

Clockwise Mode

When Pin 10 is HIGH and Pin 12 is LOW, the motor is in clockwise direction. However, Pin 11 is needed to be HIGH in order for the H-bridge circuit to operate

Anit-Clockwise Mode

When Pin 10 is LOW and Pin 12 is HIGH, the motor is in Anti-clockwise direction. Pin 11 is needed to be HIGH in order for the H-bridge circuit to operate. The L298N chip has a limitation current of 4 A to the driving motor

Stop Mode.

When Pin 10 is HIGH and Pin 12 is High, the motor power to the motor would be cut off. The stop mode can also be activated by setting a LOW input to the Pin 11 (pulse width modulation input)

Pulse Width Modulation

The pulse width modulation is used to control the duty cycle of the motor, in which the percentage of the driving output is HIGH in a frequency cycle. For example, if the duty cycle is set to be 100%, the motor is operating on maximum power thus, rotating at maximum speed. If the duty cycle is set to be 50%, the switching circuit would be operating only 50% of the time, thus the speed of the motor would be reduced.

4.7 Voltage regulation

The L7805CV voltage regulator is used to regulate to a voltage of 5V to the logic circuits. The Chip can receive up to 38V of input source. However, the output of the chip would only be 5V at maximum. This is useful to ensure the safety of circuits such as the Microcontrollers and logic gates.

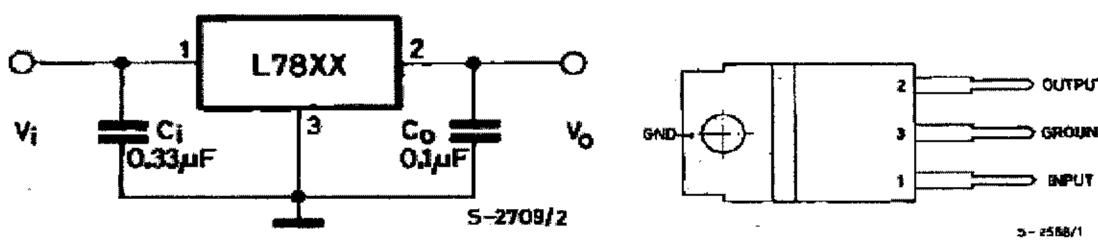


Figure 63 The Pin layout and application Diagram of the L7805 Voltage regulator

Both the input and output share a common ground. The chip is easily heated up and there would be some power loss to the environment. Therefore, to ensure the safety of the chip, heat-sinks are attached to the chip for heat dissipation. The voltage regulators are soldered on a veraboard equipped with heat-sink and switches to turn on and off anytime.

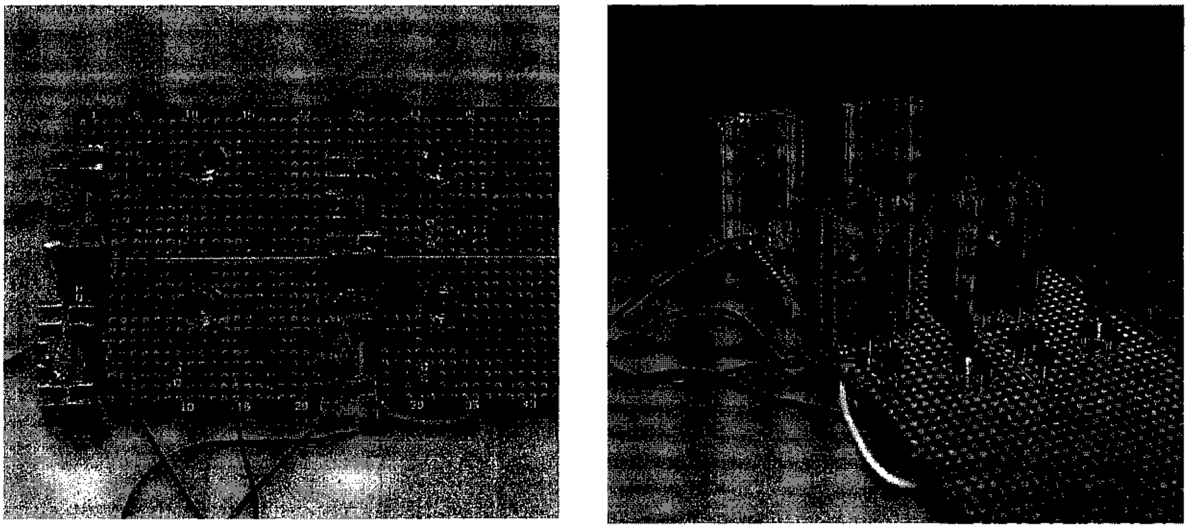


Figure 64 The L7805CV Voltage regulator equipped with heat sink and switches for application

4.8 The grass Cutter

The grass cutter for the lawn mower robot is modified from the BOSH AGS10 Cordless grass shear. The AGS10 grass shear is sawed into parts, separating the battery power supply and the cutting blades. The blades were attached at the bottom of the base of the mobile robot. The switch was modified in which it is triggered by the relay when the microcontroller gives a HIGH signal. The AGS10 operates on a 4.8V motor and has a runtime of 45 minutes when the battery is fully charged for 20 hours.

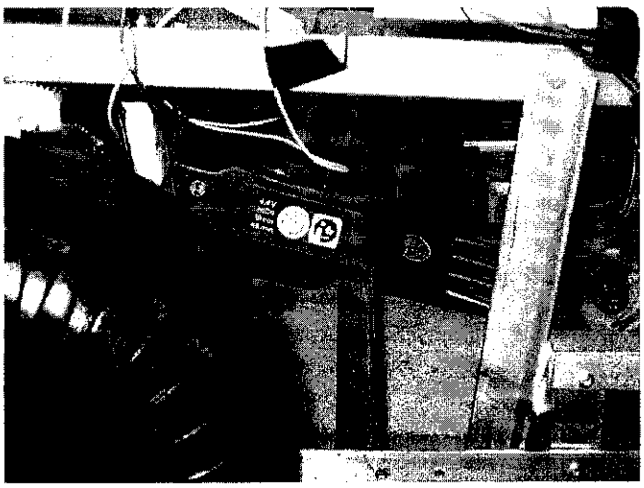


Figure 65 AGS10 grass shear attached to the lower compartment of the robot. Almost touching the ground for effective grass cutting purposes.

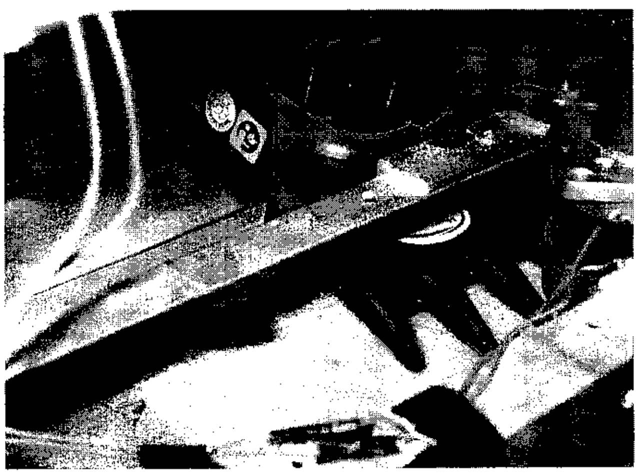
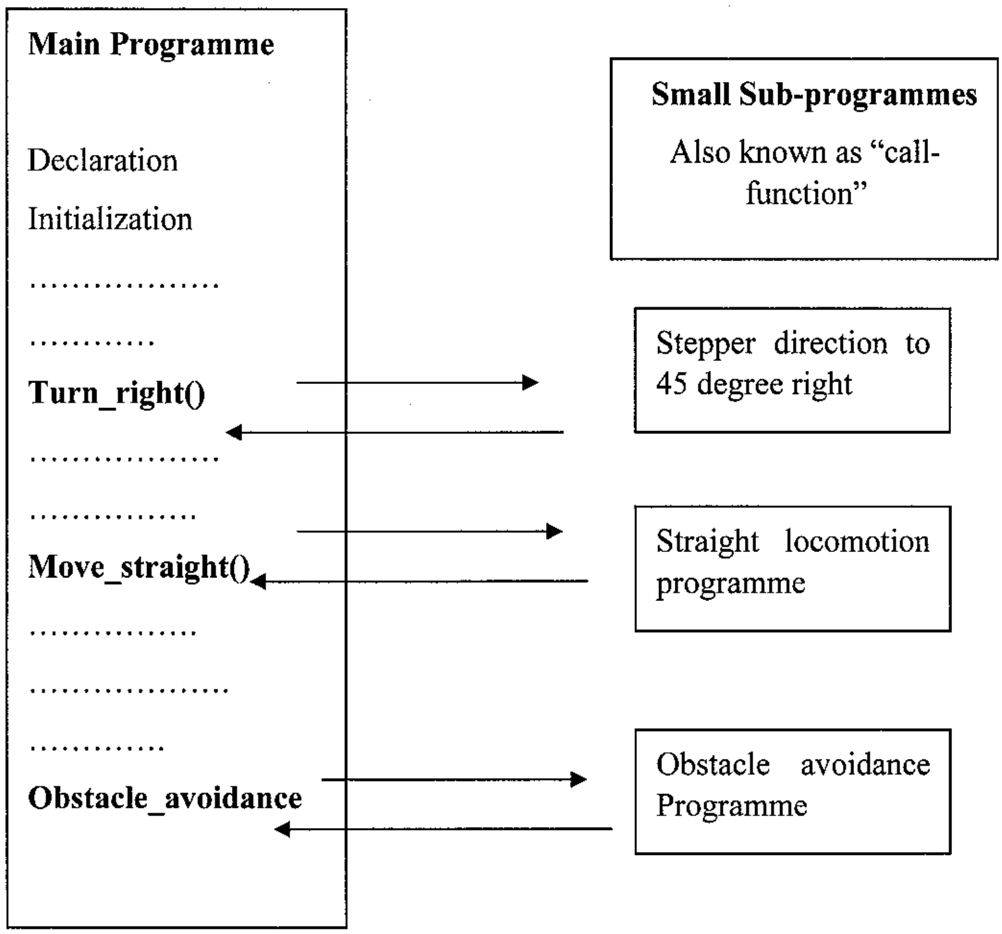


Figure 66 Close view of the grass cutter, which uses the scissor method for grass cutting. Both blades move in the opposite directions when the motor rotates.

4.9 PIC PROGRAMMING

4.9.1 Programming the PIC 16F877 Microcontroller

The C++ Programming language is use to programme the PIC microcontroller. For better strategy and organization of the programming codes, the programme is divided into small sub-programmes.



The main program would be executed when the power is turned on. The program would begin with initialization of variables and declaration. The program would then execute the instruction accordingly. The sub-program are declared with the name, **Turn_ritght()**, **Move_straight()** and **Obstacle_avoidance()**. When the main program reaches to any of the lines with the same title as these sub-programs, the PIC microcontroller would then execute an interrupt and the particular sub-program is executed. The microcontroller would then return to the main program when the execution of the sub-program is finished. The program would then continue to execute until it reaches another sub-program or ends.

4.9.2 Main Program explaination of Codes

The program functions are explained below. Please refer to Appendix A for the complete program.

/*---*/

SECTION 2

Main program.

This is the main program in which the robot would execute. The program runs by executing other small programs called "Call Functions". The program will jump to the small program and execute. When the small program finishes, it will then jump back to the main program where it left off.

/*---*/

```
void main()
{
 set_tris_a(0xff);
 set_tris_b(0x00);
 set_tris_c(0x00);
  set_tris_d(0xff);
```

Declaration of Input/Outputs

declaring all the PORTS as either inputs or outputs

```
/* this is to declare the pulse width modulation to be used for the robot*/
setup_timer_2(T2_DIV_BY_1,99,1); //enable Timer2, PR2=99, prescaler=1
setup_ccp1(CCP_PWM); //enable PWM mode
setup_ccp2(CCP_PWM); //enable PWM mode
```

Pulse Width Modulation Declaration

-to determine the value of frequency in which the PWM would operate

```
 duty_cycle = 100;
 duty_cycle2 = 100;
 CCP_1 = duty_cycle;
 CCP_2 = duty_cycle2;
 output_bit(PIN_C1,1);
 output_bit(PIN_C2,1);
```

Output of PWM

-the PWM is set to be 100%

-operating at maximum speed and power

```
output_low(PIN_B6);
output_high(PIN_B5);
delay_ms(3000);
output_high(PIN_B4);
delay_ms(30);
output_low(PIN_B4);
delay_ms(25);
```

// this small section is to configure the stepper

//for accuracy to move in a straight line

Stepper motor initial adjustment

-to correct the angle of the stepper motor in order to move straight.

```
delay_ms(2000);                  /*delay for 2s*/

movefrontovs(400,0);  --->  [example of small Sub-Programme (would jump to
                             the mini programme)]

delay_ms(1000);                  /*delay for 1s*/
turnright(76);                   /*turn right*/
delay_ms(1000);                  /*delay for 1s*/

movefrontR(20,0);                /*move front with obstacle avoidance*/
delay_ms(1000);                  /*delay for 1s*/
wheelreverse(40,0);
delay_ms(1000);                  /*delay for 1s*/

turnright(76);                   /*turn right*/
delay_ms(1000);                  /*delay for 10s*/

movefrontovsR(400,0);
delay_ms(1000);                  /*delay for 10s*/

turnleft(76);                    /*turn left*/
delay_ms(1000);                  /*delay for 10s*/

movefrontL(20,0);                /*move front without ovs*/
delay_ms(1000);                  /*delay for 10s*/
wheelreverse(40,0);
delay_ms(1000);

turnleft(76);                    /*turn left*/
delay_ms(1000);                  /*delay for 10s*/
```

Programme using small sub-programmes

-the programme would execute other programmes with the title such as **movefrontR()** and would return to the main programme when all the instructions are executed

4.9.3 Example of Subsection Codes used for the Robot

```
/*-----------------------------------------------------------------------------------------
SECTION 4
FORWARD LOCOMOTION WITH COLLISION AVOIDANCE
this programme is used for moving the robot to the front. It includes the OVS (obstacle avoidance system)
therefore, if there is a signal in the ultrasonic sensor, the obstacleavoid programme will execute

Pin_A1= encoder for the right wheel
Pin_A2=ultrasonic sensor
-----------------------------------------------------------------------------------------*/

void wheelfrontovs(int m,int n)

{

int pulse;
int count;
count=n;

        while(true)
        {

                pulse=!input(PIN_D2);
                while(pulse==!input(PIN_D2))
                        {
        CCP_1=duty_cycle;
        CCP_2=duty_cycle2;

        duty_cycle = 100;
        duty_cycle2 = 100;

                        output_high(PIN_B0);
                        output_low(PIN_B1);
                        output_high(PIN_B2);
                        output_low(PIN_B3);
                        }
                count=count+1;
```

Declaration of Variables Used

-pulse is used to set the input from encoder

-count is needed to count the number of pulses already passed (distance already traveled)

Wheel Encoder receiving LOW input

-input from the distance encoder

The locomotion moves forward

As long as it does not receive the number of pulses needed (the traveled distance have not been met)

```
if(!input(PIN_D3))
      {
   wheelreverse(40,0);   // reverse locomotion for a short distance
   delay_ms(1000);
   turnleft(76);         //forward locomotion
   movefrontL(100,0);
   turnright(76);        //forward locomotion
   movefrontR(140,0);
   turnright(76);        //forward locomotion
   movefrontR(100,0);
   turnleft(76);
   centerfromleft();

            count=count+100;
   }

            if(count>m-1)
                  {
                  break;
                  }


 }

output_high(PIN_B0);    /*h-bridge A stop*/
output_high(PIN_B1);
output_high(PIN_B2);    /*h-bridge B stop*/
output_high(PIN_B3);
return;

}
```

When an input Low is received in D3

The obstacle is detected by the ultrasonic sensor

Obstacle Avoidance Programme

-The robot would reverse 40 cm

-turn left and go around the obstacle

Distance Compensation Algorithm

-The original path planned must deduct the distance traveled during the obstacle avoidance programme

-The Sub-programme has finished execution and would return to the main programme from this point to where it has left off.

CHAPTER 5
CONCLUSION

The project objectives have been successfully met as the robot is able to move according to the predetermined path defined by the user while avoiding any obstacle blocking its path. The robot structure is constructed and able to move. Interaction between sensors, microcontroller and output drives are working and communicating to each other accordingly.

To further improve the performance, these are some recommendations suggested:

- The geared motor is accurate but slow however, a better and faster geared DC motor is needed to further improve the speed of the robot
- The ground of the lawn is not totally flat; therefore, balancing sensors are suggested to improve the robot to compensate for the difference in distance traveled between the two wheels for smooth straight motion path.
- The robot is working under the sun, a solar panel can be suggested to power up the battery and power up the circuitries such as the microcontroller and logic circuits to reduce the burden of the battery.
- A more sophisticated and user friendly program can be developed for practical usage of the robot.

REFERENCES

1. Groover, Mikell.P. 1986, *Industrial Robotics: Technology, Programming and Applications*, USA, McGraw-Hill.

2. Iovine, John. 2001, *Robots, Androids and Animatrons; 2nd Edition*, USA, McGraw-Hill..

3. Asfahl, C. Ray, 1992, *Robots and Manufacturing Automation*, Canada, John Wiley & Sons, Inc.

4. Joseph L.Jones, Anita M .Flynn, Bruce A.Seiger, *MOBILE ROBOTS: Inspiration to Implimentation*; *Second Edition*, A K Peters Natick, Massachusetts.

5. Jorge Angeles, Fundamentals of Robotic Mechanical System, Canada, John Wiley & Sons, Inc.

6. Richard Barnett, Sarah Cox & Larry O' Cull, Embedded C Programming and the Microchip PIC, Canada, Delmar Learning, Thompson Learning.

7. http://en.wikipedia.org

8. McComb, Gordon. Robot Builder's Bonanaza: 99 Inexpensive Robotic Projects, USA, McGraw-Hill.

9. http://www.beam-online.com/navagation/_info_tut.htm

10. http://hypertextbook.com/physics/mechanics/friction/

APPENDICES

APPENDIX A
PIC PROGRAMMING CODES

Appendix A:1 Main program for the !6F877PIC controller with obstacle avoidance

```
/*-----------------------------------------------------------------------------------------------------------
SECTION 1
 This programme is written to move a straight line and back
it does not move in a box and the distance for the path is already determined*/
/*---------------------------------------------------------------------------------------------------------*/

#include <16f877.h>
#USE DELAY(CLOCK = 4000000)
#include <stdio.h>
#include <stdlib.h>
#FUSES XT,NOWDT, NOPROTECT, NOPUT,  NOBROWNOUT, NOLVP


void wheelfrontovs(int m , int n);                    /*declaration of call function*/
void wheelfront(int m, int n);
void wheelright(int m);
void wheelleft(int m);
void wheelreverse(int m, int n);

void centerfromleft();
void centerfromright();
void stepperleft();
void stepperright();
void turnleft(int m);
void turnright(int m);

void movefrontovs(int m, int n);
void movefrontovsL(int m, int n);
void movefrontovsR(int m, int n);
void movefront(int m,int n);
void movefrontL(int m,int n);
void movefrontR(int m,int n);
```

```
void centerfromleft(void);
void centerfromright(void);



int i;                                  //declaration of representation for value input
int duty_cycle;
int duty_cycle2;
```

/*---

This is just for reference of distance for unit pulse. The calculation provided input to the controller about the distance traveled by the robot. Extremely useful for path planning.

<u>Distance calculation</u>

Distance for single pulse = 2*pi*j/60

j = 9.75cm

Therefore, 1 pulse = section = 1 cm

1 rotation = 60 cm traveled by robot

---*/

/*---*/

SECTION 2

Main programme.

This is the main program in which the robot would execute. The program runs by executing other small programs called "Call Functions". The program will jump to the small program and execute. When the small program finishes, it will then jump back to the main program where it left off.

/*---*/

```
void main()
{
 set_tris_a(0xff);                      //declaring all the PORTS as either inputs or outputs
 set_tris_b(0x00);
 set_tris_c(0x00);
  set_tris_d(0xff);
```

```
/* this is to declare the pulse width modulation to be used for the robot*/
setup_timer_2(T2_DIV_BY_1,99,1); //enable Timer2, PR2=99, prescaler=1
setup_ccp1(CCP_PWM); //enable PWM mode
setup_ccp2(CCP_PWM); //enable PWM mode


 duty_cycle = 100;
 duty_cycle2 = 100;
 CCP_1 = duty_cycle;
 CCP_2 = duty_cycle2;
 output_bit(PIN_C1,1);
 output_bit(PIN_C2,1);

output_low(PIN_B7);                     // turn off the grass cutter

output_low(PIN_B6);                     // this small section is to configure the stepper
output_high(PIN_B5);                    //for accuracy to move in a straight line
delay_ms(3000);
output_high(PIN_B4);
delay_ms(30);
output_low(PIN_B4);
delay_ms(25);


                              //beginning of the main program execution instructions
delay_ms(2000);               /*delay for 2s*/
movefrontovs(400,0);          /*move front with obstacle avoidance*/
delay_ms(1000);               /*delay for 10s*/


turnright(76);                /*turn right*/
delay_ms(1000);               /*delay for 10s*/

movefrontR(20,0);             /*move front with obstacle avoidance*/
delay_ms(1000);               /*delay for 10s*/
wheelreverse(40,0);
delay_ms(1000);               /*delay for 10s*/

turnright(76);                /*turn right*/
delay_ms(1000);               /*delay for 10s*/
```

```
movefrontovsR(400,0);                /*move front with obstacle avoidance*/
delay_ms(1000);                      /*delay for 10s*/

turnleft(76);                        /*turn left*/
delay_ms(1000);                      /*delay for 10s*/

movefrontL(20,0);                    /*move front without ovs*/
delay_ms(1000);                      /*delay for 10s*/
wheelreverse(40,0);
delay_ms(1000);

turnleft(76);                        /*turn left*/
delay_ms(1000);                      /*delay for 10s*/

}

/*-----------------------------------------------------------------------
SECTION 3
Stepper motor direction Control using "Call Functions"
 this programme is to control the stepper motor. pulsing it to be at the centre
Pin_B4 = pulsing clock
Pin_B5 = clockwise/anticlockwise  assume high =clockwise
Pin_B6 = full/half                assume high = full
-----------------------------------------------------------------------*/

void centerfromleft(void)
{
output_low(PIN_B6);
output_high(PIN_B5);

//int i;
for(i=0; i<25; i++)
{
output_high(PIN_B4);
delay_ms(30);
output_low(PIN_B4);
delay_ms(25);
}
return;
}
```

```
void centerfromright(void)
{
output_low(PIN_B6);
output_low(PIN_B5);

//int i;
for(i=0; i<25; i++)
{
output_high(PIN_B4);
delay_ms(30);
output_low(PIN_B4);
delay_ms(25);
}
return;
}

void stepperleft(void)
{
output_low(PIN_B6);
output_low(PIN_B5);

//int i;
for(i=0;i<25;i++)
{
output_high(PIN_B4);
delay_ms(30);
output_low(PIN_B4);
delay_ms(25);
}
return;
}

void stepperright(void)
{
output_low(PIN_B6);
output_high(PIN_B5);

//int i;
for(i=0;i<25;i++)
```

```
{
output_high(PIN_B4);
delay_ms(30);
output_low(PIN_B4);
delay_ms(25);
}
return;
}


/*--------------------------------------------------------------------------------------

SECTION 4
FORWARD LOCOMOTION WITH COLLISION AVOIDANCE
this programme is used for moving the robot to the front. It includes the OVS (obstacle avoidance
system)
therefore, if there is a signal in the ultrasonic sensor, the obstacleavoid programme will execute

Pin_A1= encoder for the right wheel
Pin_A2=ultrasonic sensor
----------------------------------------------------------------------------------------*/

void wheelfrontovs(int m,int n)

{

int pulse;
int count;
int var=0;
count=n;

          while(true)
          {

output_high(PIN_B7);                                //turn on the grass cutter as it moves forward

                    pulse=!input(PIN_D2);
                    while(pulse==!input(PIN_D2))
                              {
           CCP_1=duty_cycle;
          CCP_2=duty_cycle2;
```

```
        duty_cycle = 100;
        duty_cycle2 = 100;

                        output_high(PIN_B0);                /*move h-bridge A forward*/
                        output_low(PIN_B1);
                        output_high(PIN_B2);                /*move h-bridge B forward*/
                        output_low(PIN_B3);
                        }
                count=count+1;

    if(!input(PIN_D1))                      //if the PIN_D3 goes low, which is the Ultrasonic Sensor
        {

output_low(PIN_B7);                         //turn off grass cutter
output_high(PIN_B0);                        /*h-bridge A stop*/
output_high(PIN_B1);
output_high(PIN_B2);                        /*h-bridge B stop*/
output_high(PIN_B3);
delay_ms(1000);


    wheelreverse(40,0);             // reverse locomotion for a short distance
    delay_ms(1000);
    turnleft(76);                   //forward locomotion
    movefrontL(100,0);
    turnright(76);                  //forward locomotion
    movefrontR(140,0);
    turnright(76);                  //forward locomotion
    movefrontR(100,0);
    turnleft(76);
    centerfromleft();
                count=count+100;
      }

                if(count>m-1)
                        {
                        break;
                        }
```

```
  }
output_low(PIN_B7);                    //turn off grass cutter
output_high(PIN_B0);                   /*h-bridge A stop*/
output_high(PIN_B1);
output_high(PIN_B2);                   /*h-bridge B stop*/
output_high(PIN_B3);
return;

}




/*-------------------------------------------------------------------------------------
SECTION 5
FORWARD LOCOMOTION WITHOUT OBSTACLE AVOIDANCE
this programme is used for moving the robot to the front. It does not include the obstacle avoidance
system
Pin_A0= encoder for the left wheel
Pin_A1= encoder for the right wheel
-------------------------------------------------------------------------------------*/

void wheelfront(int m,int n)

{
int pulse;
int count;
count=n;

        while(true)
        {

                output_high(PIN_B7);            //turn on the grass cutter

                pulse=!input(PIN_D2);
                while(pulse==!input(PIN_D2))
                        {
        CCP_1=duty_cycle;
        CCP_2=duty_cycle2;
        duty_cycle = 100;
         duty_cycle2 = 100;
```

```
                    output_high(PIN_B0);                    //move h-bridge A forward
                    output_low(PIN_B1);
                    output_high(PIN_B2);                    //move h-bridge B forward
                    output_low(PIN_B3);
                    }
            count=count+1;

            if(count>m-1)
                    {
                    break;
                    }


        }
  output_low(PIN_B7);                    //turn off grass cutter

output_high(PIN_B0);                     //h-bridge A stop
output_high(PIN_B1);
output_high(PIN_B2);                     // h-bridge B stop
output_high(PIN_B3);
return;
}


/*-------------------------------------------------------------------------------------
SECTION 6
LEFT LOCOMOTION
this programme is used for moving the robot to the left. It does not include the obstacle avoidance
system
Pin_A0= encoder for the left wheel
Pin_A1= encoder for the right wheel
-------------------------------------------------------------------------------------*/

void wheelleft(int m)

{

int pulse;
int count;
count=0;
```

```
while(true)
{

        pulse=!input(PIN_D2);           //checking for the low input of encoder
        while(pulse==!input(PIN_D2))    //every low input of encoder would give a count
                {
    CCP_1=duty_cycle;
    CCP_2=duty_cycle2;
                duty_cycle = 100;
                duty_cycle2 = 100;

                output_low(PIN_B0);             /*stop the left wheel*/
                output_low(PIN_B1);
                output_high(PIN_B2);            /*move h-bridge B forward*/
                output_low(PIN_B3);
                }
        count=count+1;

        if(count>m-1)
                {
                break;
                }


}
output_high(PIN_B0);            /*h-bridge A stop*/
output_high(PIN_B1);
output_high(PIN_B2);            /*h-bridge B stop*/
output_high(PIN_B3);
return;
}


/*-------------------------------------------------------------------------------------
SECTION 7
RIGHT LOCOMOTION
this programme is used for moving the robot to the right. It does not include the obstacle avoidance
system
Pin_A0= encoder for the left wheel
Pin_A1= encoder for the right wheel
-------------------------------------------------------------------------------------*/
```

```
void wheelright(int m)

{

int pulse;
int count;
count=0;

          while(true)
          {

                    pulse=!input(PIN_D0);
                    while(pulse==!input(PIN_D0))
                              {

      CCP_1=duty_cycle;
      CCP_2=duty_cycle2;
                              duty_cycle = 100;
                              duty_cycle2 = 100;

                              output_high(PIN_B0);                  /*move h-bridge A forward*/
                              output_low(PIN_B1);
                              output_high(PIN_B2);                  /*stop the right wheel*/
                              output_high(PIN_B3);
                              }
                    count=count+1;

                    if(count>m-1)
                              {
                              break;
                              }


          }
output_high(PIN_B0);                  /*h-bridge A stop*/
output_high(PIN_B1);
output_high(PIN_B2);                  /*h-bridge B stop*/
output_high(PIN_B3);
return;
}
```

```
/*----------------------------------------------------------------------------------------
SECTION 8
Reverse LOCOMOTION
this programme is used for moving the robot to the reverse direction. It does not include the obstacle
avoidance system
Pin_A0= encoder for the left wheel
Pin_A1= encoder for the right wheel
----------------------------------------------------------------------------------------*/

void wheelreverse(int m, int n)

{

int pulse;
int count;
count=n;

          while(true)
          {

                    pulse=!input(PIN_D2);
                    while(pulse==!input(PIN_D2))
                              {
                              CCP_1=duty_cycle;
                               CCP_2=duty_cycle2;
                              duty_cycle = 100;
                               duty_cycle2 = 100;

                              output_low(PIN_B0);                    /*move h-bridge A reverse*/
                              output_high(PIN_B1);
                              output_low(PIN_B2);                    /*move h-bridge A reverse*/
                              output_high(PIN_B3);
                              }
                    count=count+1;

                    if(count>m-1)
                              {
                              break;
                              }
```

```
        }
output_high(PIN_B0);                    /*h-bridge A stop*/
output_high(PIN_B1);
output_high(PIN_B2);                    /*h-bridge B stop*/
output_high(PIN_B3);
return;
}
```

```
/*--------------------------------------------------------------------------------------------
SECTION 9
LEFT DIRECTION
this programme is used for controlling the direction for the stepper motor. To move it to the
Left. The concept just combines both direction control sub programme and the locomotion sub
programme to be as a bigger programme.
----------------------------------------------------------------------------------------------*/

void turnleft(int m)
{
delay_ms(1000);
stepperleft();          //left stepper direction
delay_ms(1000);
wheelleft(m);           //left locomotion
delay_ms(1000);
return;
}
```

```
/*--------------------------------------------------------------------------------------------
SECTION 10
RIGHT DIRECTION
this programme is used for controlling the direction for the stepper motor. To move it to the
right. The concept just combines both direction control sub programme and the locomotion sub
programme to be as a bigger programme.
----------------------------------------------------------------------------------------------*/
```

```
void turnright(int m)
{
delay_ms(1000);
stepperright();           //right stepper direction
delay_ms(1000);
wheelright(m);            //right locomotion
delay_ms(1000);
return;
}



/*-------------------------------------------------------------------------------------
SECTION 11
Forward DIRECTION WITHOUT RECOVERY
this programme is the combination of the stepper direction and the locomotion for moving forward
The concept just combines both direction control sub programme and the locomotion sub programme
to be as a bigger programme
-------------------------------------------------------------------------------------*/


void movefrontovs(int m, int n)
{

delay_ms(1000);
wheelfrontovs(m,n);                 //Forward locomotion
delay_ms(1000);
return;
}



/*-------------------------------------------------------------------------------------
SECTION 12
Forward DIRECTION FROM LEFT
this programme is the combination of the stepper direction and the locomotion for moving forward
RECOVERING FROM LEFT. The concept just combines both direction control sub programme and
the locomotion sub programme to be as a bigger programme.
-------------------------------------------------------------------------------------*/
```

```
void movefrontovsL(int m, int n)
{
delay_ms(1000);
centerfromleft();           // recover from left turning position
delay_ms(1000);
wheelfrontovs(m,n);                 //forward locomotion
delay_ms(1000);
return;
}
```

```
/*------------------------------------------------------------------------------------------
SECTION 13
Forward DIRECTION FROM RIGHT
this programme is the combination of the stepper direction and the locomotion for moving forward
RECOVERING FROM RIGHT. The concept just combines both direction control sub programme and
the locomotion sub programme to be as a bigger programme
------------------------------------------------------------------------------------------*/
```

```
void movefrontovsR(int m, int n)
{
delay_ms(1000);
centerfromright();           // recover from right turning position
delay_ms(1000);
wheelfrontovs(m,n);                 //forward locomotion
delay_ms(1000);
return;
}
```

```
/*------------------------------------------------------------------------------------------
SECTION 14
Forward DIRECTION WITHOUT RECOVERY AND OVS
this programme is the combination of the stepper direction and the locomotion for moving forward
The concept just combines both direction control sub programme and the locomotion sub programme
```

```
to be as a bigger programme
------------------------------------------------------------------------------------------*/


void movefront(int m, int n)
{

delay_ms(1000);
wheelfront(m,n);            //forward locomotion
delay_ms(1000);
return;
}


/*------------------------------------------------------------------------------------------
SECTION 15
Forward DIRECTION FROM LEFT WITHOUT OVS
this programme is the combination of the stepper direction and the locomotion for moving forward
RECOVERING FROM LEFT. The concept just combines both direction control sub programme and
the locomotion sub programme to be as a bigger programme
------------------------------------------------------------------------------------------*/


void movefrontL(int m, int n)
{
delay_ms(1000);
centerfromleft();           // recover from left turning position
delay_ms(1000);
wheelfront(m,n);            //forward locomotion
delay_ms(1000);
return;
}


/*------------------------------------------------------------------------------------------
SECTION 16
Forward DIRECTION FROM RIGHT WITHOUT OVS
this programme is the combination of the stepper direction and the locomotion for moving forward
RECOVERING FROM RIGHT. The concept just combines both direction control sub programme and
```

the locomotion sub programme to be as a bigger programme

--*/

```
void movefrontR(int m, int n)
{
delay_ms(1000);
centerfromright();                    // recover from right turning position
delay_ms(1000);
wheelfront(m,n);            //forward locomotion
delay_ms(1000);
return;
}
```

APPENDIX B

COEFFICIENT OF FRICTION FORCE

Coefficients of Friction for Selected Interfaces
(in order of generally decreasing value)

μ_s	μ_k	Interface
	1.16	rubber - rubber
	1.02	rubber - concrete
	0.72	car tire - asphalt
	0.35	car tire - grass
0.8 - 1.0		skin - metals
0.9 - 1.0		glass - glass
	0.9	sheep - steel mesh
	0.7	sheep - plastic batten (□)
	0.6	sheep - plastic batten (□)
	0.6	sheep - wood batten (□)
	0.5	sheep - wood batten (□)
0.58		steel - steel
0.4		brakes - cast iron
0.6		wood - brick
0.2 - 0.6		wood - metals
0.29	0.22	wood - felt
0.28	0.17	wood - wood
0.3		snow - nylon
0.04 - 0.4	0.04 - 0.4	snow - hickory, waxed
0.1		graphite - graphite
0.1		graphite - steel
	0.03	ice - steel
0.05 - 0.5	0.02 - 0.09	ice - ice
0.2		teflon - steel
0.04		teflon - teflon
	0.0013	tendon - sheath
	0.003	normal bone joints

APPENDIX C

PIC 16F877 DATASHEET

MICROCHIP

PIC16F87X Data Sheet

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

DNV Certification, Inc. USA

DNV MSC The Netherlands Accredited by the RvA

ISO 9001 / QS-9000 REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

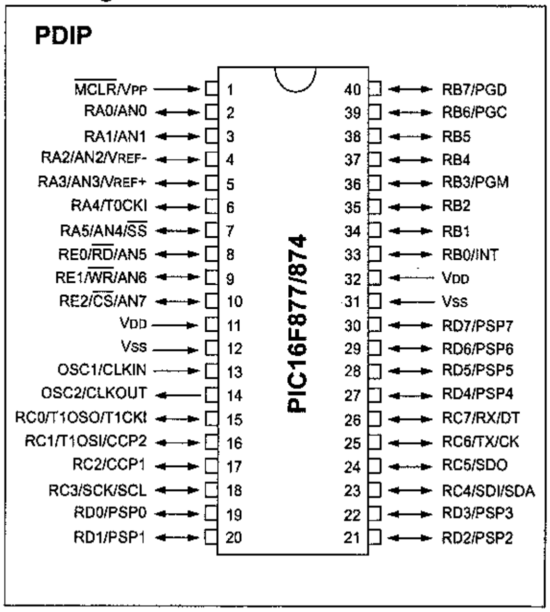
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
  Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

Pin Diagram

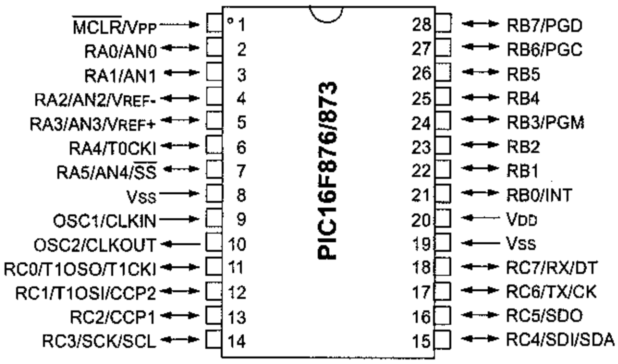


Peripheral Features:

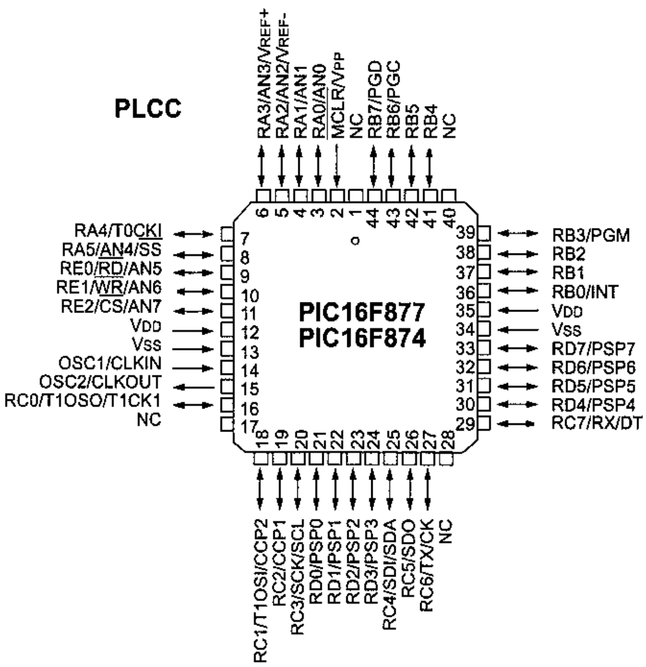
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max.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

Pin Diagrams

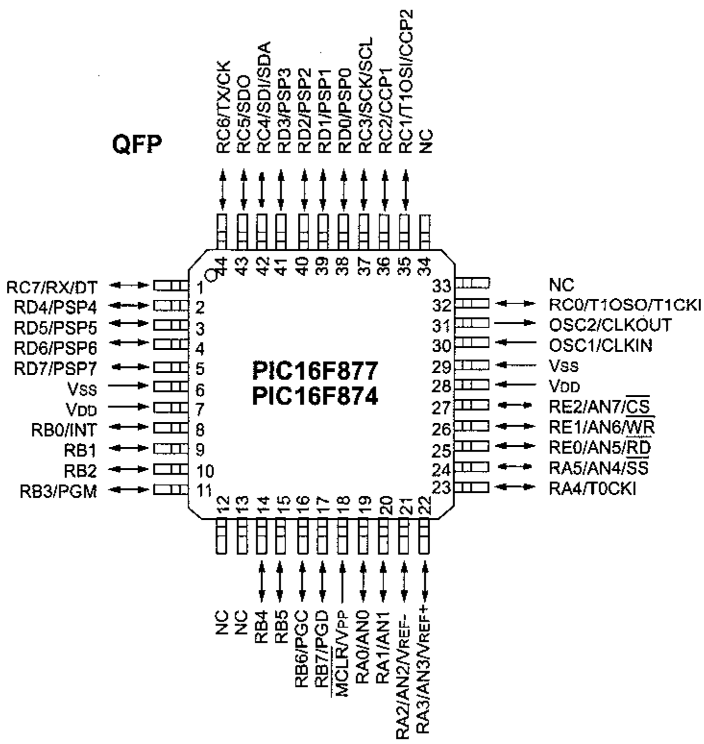
PDIP, SOIC



PLCC



QFP



Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	**PIC16F873**	**PIC16F874**	**PIC16F876**	**PIC16F877**
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com/cn** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

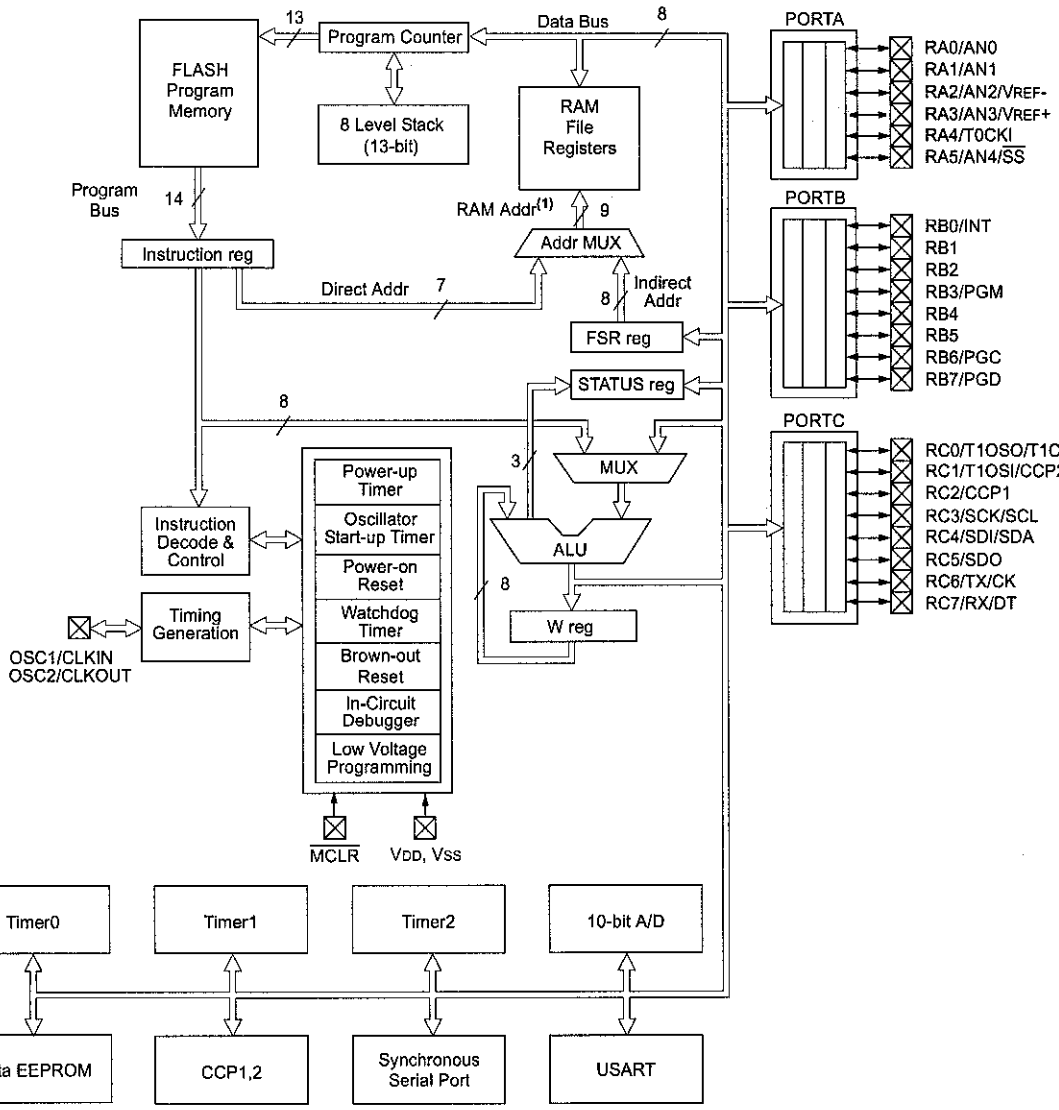
This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The Parallel Slave Port is not implemented on the 28-pin devices.

The following device block diagrams are sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM

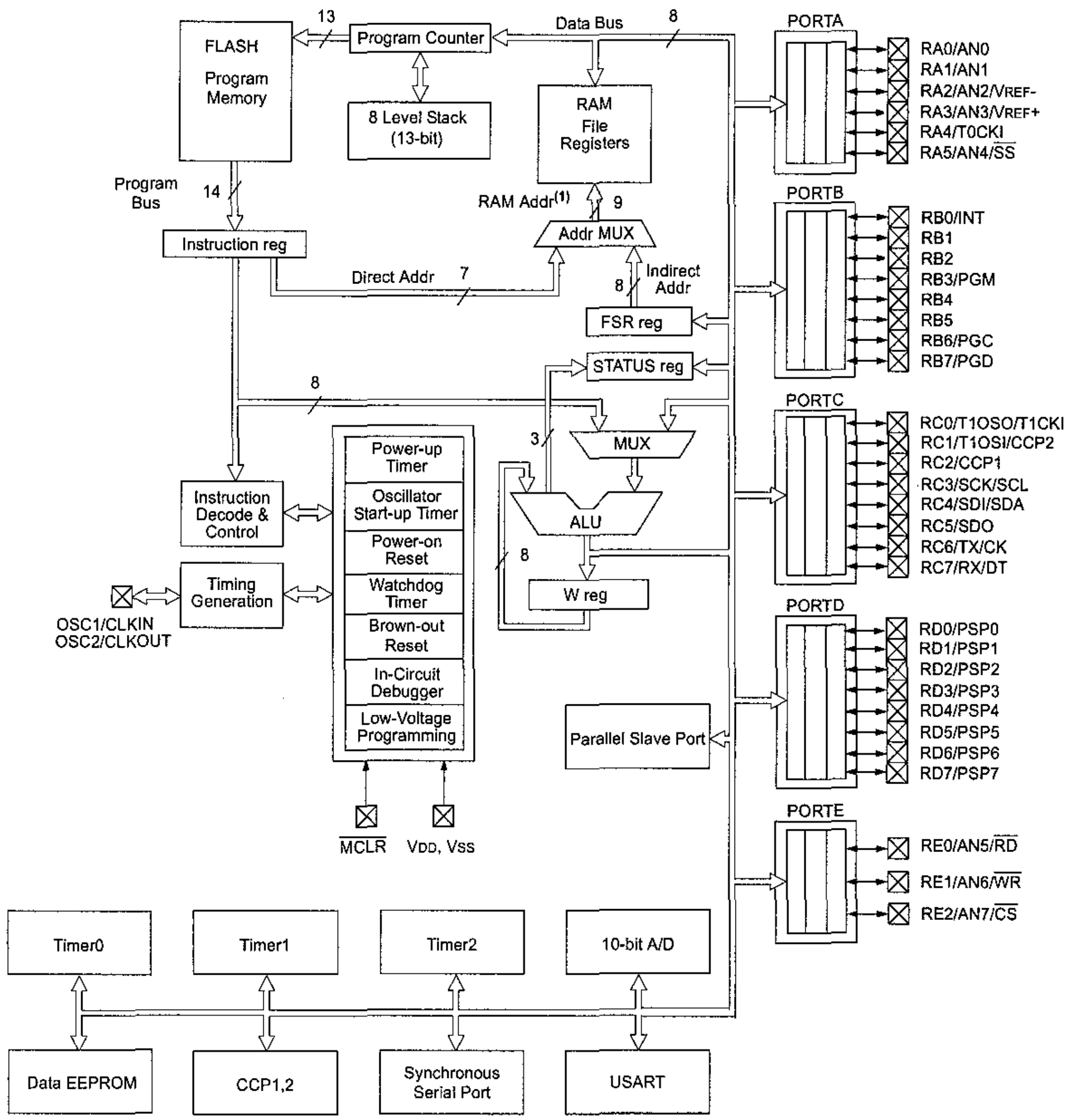
Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$/VPP	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	2	2	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/$\overline{\text{SS}}$/AN4	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	25	25	I/O	TTL	Interrupt-on-change pin.
RB5	26	26	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/ PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS(4)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/ counter. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	37	41	14	I/O	TTL	Interrupt-on-change pin.
RB5	38	42	15	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	39	43	16	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

APPENDIX D

LM78XX VOLTAGE REGULATOR DATASHEET

National Semiconductor

May 2000

LM78XX
Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expanded to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

Features

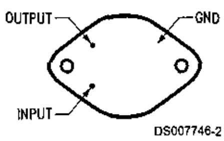
- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

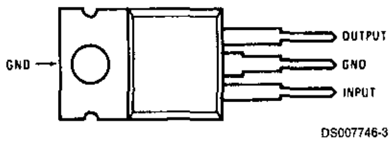
Connection Diagrams

Metal Can Package
TO-3 (K)
Aluminum



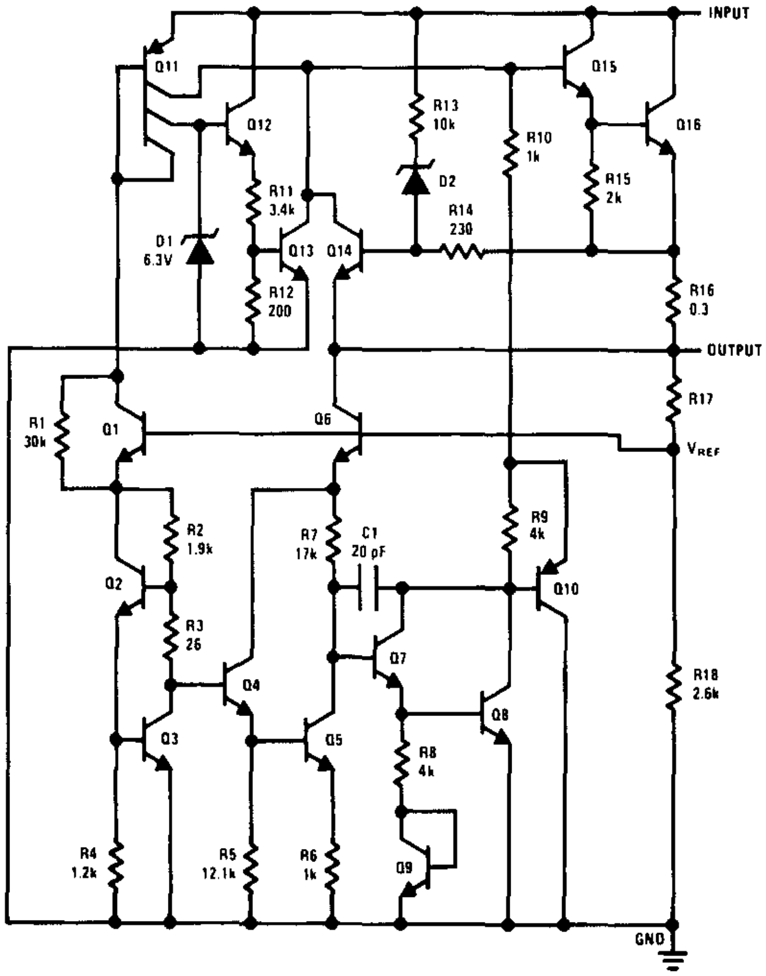
Bottom View
Order Number LM7805CK,
LM7812CK or LM7815CK
See NS Package Number KC02A

Plastic Package
TO-220 (T)



Top View
Order Number LM7805CT,
LM7812CT or LM7815CT
See NS Package Number T03B

Schematic



DS007746-1

ɔsolute Maximum Ratings (Note 3)

Military/Aerospace specified devices are required, ase contact the National Semiconductor Sales Office/ ıtributors for availability and specifications.

ut Voltage	
V_O = 5V, 12V and 15V)	35V
ɜrnal Power Dissipation (Note 1)	Internally Limited
erating Temperature Range (T_A)	0°C to +70°C
Maximum Junction Temperature	
(K Package)	150°C
(T Package)	150°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	
TO-3 Package K	300°C
TO-220 Package T	230°C

ectrical Characteristics LM78XXC (Note 2)

≤ T_J ≤ 125°C unless otherwise noted.

	Output Voltage			5V			12V			15V			
	Input Voltage (unless otherwise noted)			10V			19V			23V			Units
ɔl	Parameter	Conditions		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
	Output Voltage	Tj = 25°C, 5 mA ≤ I_O ≤ 1A		4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		P_D ≤ 15W, 5 mA ≤ I_O ≤ 1A		4.75		5.25	11.4		12.6	14.25		15.75	V
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7.5 ≤ V_{IN} ≤ 20)			(14.5 ≤ V_{IN} ≤ 27)			(17.5 ≤ V_{IN} ≤ 30)		V
	Line Regulation	I_O = 500 mA	Tj = 25°C		3	50		4	120		4	150	mV
			ΔV_{IN}		(7 ≤ V_{IN} ≤ 25)			14.5 ≤ V_{IN} ≤ 30)			(17.5 ≤ V_{IN} ≤ 30)		V
			0°C ≤ Tj ≤ +125°C			50			120			150	mV
			ΔV_{IN}		(8 ≤ V_{IN} ≤ 20)			(15 ≤ V_{IN} ≤ 27)			(18.5 ≤ V_{IN} ≤ 30)		V
		I_O ≤ 1A	Tj = 25°C			50			120			150	mV
			ΔV_{IN}		(7.5 ≤ V_{IN} ≤ 20)			(14.6 ≤ V_{IN} ≤ 27)			(17.7 ≤ V_{IN} ≤ 30)		V
			0°C ≤ Tj ≤ +125°C			25			60			75	mV
			ΔV_{IN}		(8 ≤ V_{IN} ≤ 12)			(16 ≤ V_{IN} ≤ 22)			(20 ≤ V_{IN} ≤ 26)		V
	Load Regulation	Tj = 25°C	5 mA ≤ I_O ≤ 1.5A		10	50		12	120		12	150	mV
			250 mA ≤ I_O ≤ 750 mA			25			60			75	mV
		5 mA ≤ I_O ≤ 1A, 0°C ≤ Tj ≤ +125°C				50			120			150	mV
	Quiescent Current	I_O ≤ 1A	Tj = 25°C			8			8			8	mA
			0°C ≤ Tj ≤ +125°C			8.5			8.5			8.5	mA
	Quiescent Current Change	5 mA ≤ I_O ≤ 1A				0.5			0.5			0.5	mA
		Tj = 25°C, I_O ≤ 1A				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7.5 ≤ V_{IN} ≤ 20)			(14.8 ≤ V_{IN}≤ 27)			(17.9 ≤ V_{IN} ≤ 30)		V
		I_O ≤ 500 mA, 0°C ≤ Tj ≤ +125°C				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7 ≤ V_{IN} ≤ 25)			(14.5 ≤ V_{IN}≤ 30)			(17.5 ≤ V_{IN} ≤ 30)		V
	Output Noise Voltage	T_A =25°C, 10 Hz ≤ f ≤ 100 kHz			40			75			90		µV
V_{IN} / OUT	Ripple Rejection	f = 120 Hz	I_O ≤ 1A, Tj = 25°C or	62	80		55	72		54	70		dB
			I_O ≤ 500 mA 0°C ≤ Tj ≤ +125°C	62			55			54			dB
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(8 ≤ V_{IN} ≤ 18)			(15 ≤ V_{IN} ≤ 25)			(18.5 ≤ V_{IN} ≤ 28.5)		V
	Dropout Voltage	Tj = 25°C, I_{OUT} = 1A			2.0			2.0			2.0		V
	Output Resistance	f = 1 kHz			8			18			19		mΩ

Electrical Characteristics LM78XXC (Note 2) (Continued)

0°C ≤ T_J ≤ 125°C unless otherwise noted.

Output Voltage			5V			12V			15V			Units
Input Voltage (unless otherwise noted)			10V			19V			23V			
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
	Short-Circuit Current	Tj = 25°C		2.1			1.5			1.2		A
	Peak Output Current	Tj = 25°C		2.4			2.4			2.4		A
	Average TC of V_{OUT}	0°C ≤ Tj ≤ +125°C, I_O = 5 mA		0.6			1.5			1.8		mV/°C
V_{IN}	Input Voltage Required to Maintain Line Regulation	Tj = 25°C, I_O ≤ 1A		7.5		14.6			17.7			V

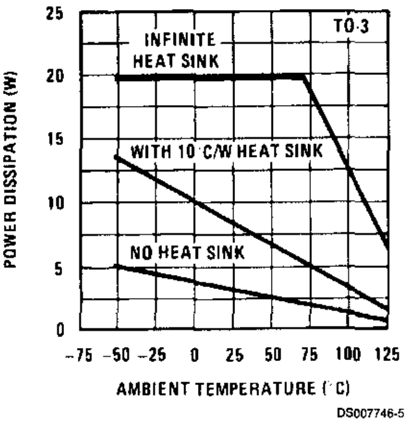
Note 1: Thermal resistance of the TO-3 package (K, KC) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 4°C/W junction to case and 50°C/W case to ambient.

Note 2: All characteristics are measured with capacitor across the input of 0.22 µF, and a capacitor across the output of 0.1µF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques (t_W ≤ 10 ms, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

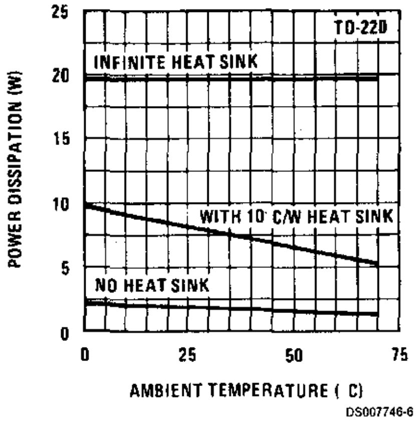
Note 3: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. For guaranteed specifications and the test conditions, see Electrical Characteristics.

ypical Performance Characteristics

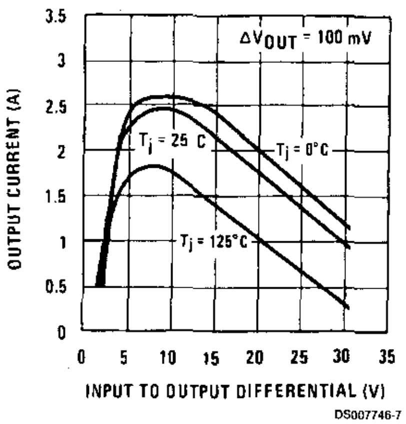
ximum Average Power Dissipation



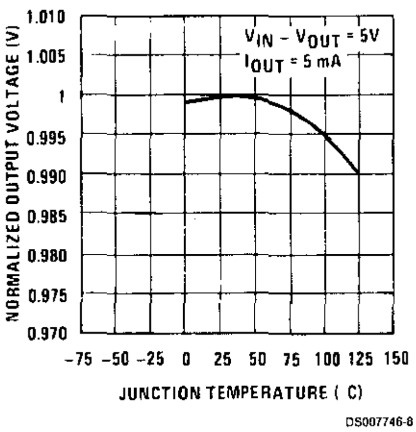
Maximum Average Power Dissipation



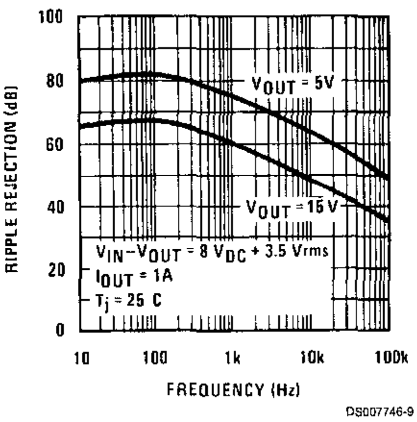
ak Output Current



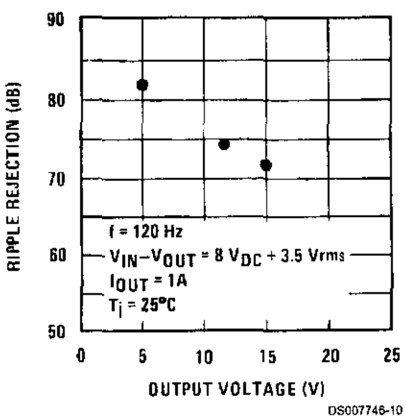
Output Voltage (Normalized to 1V at T_J = 25°C)



pple Rejection

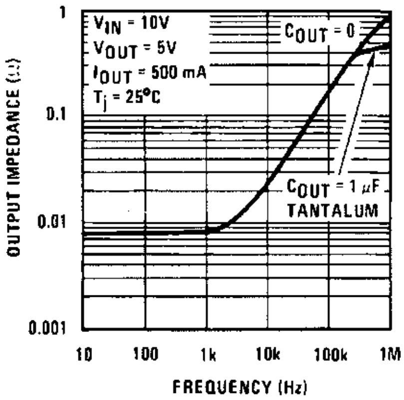


Ripple Rejection



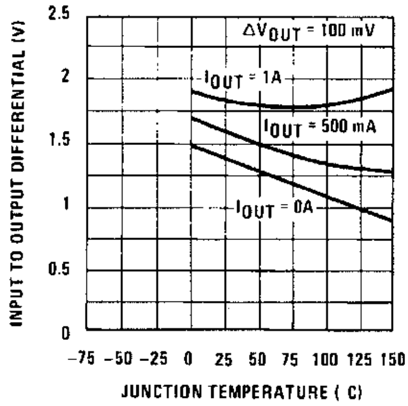
Typical Performance Characteristics (Continued)

Output Impedance



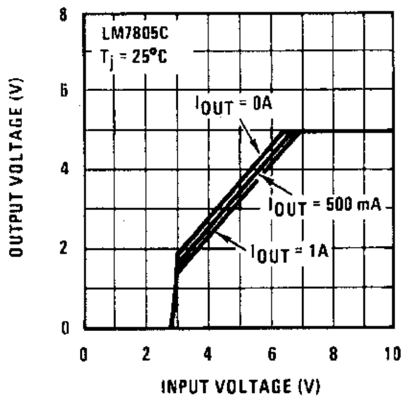
DS007746-11

Dropout Voltage



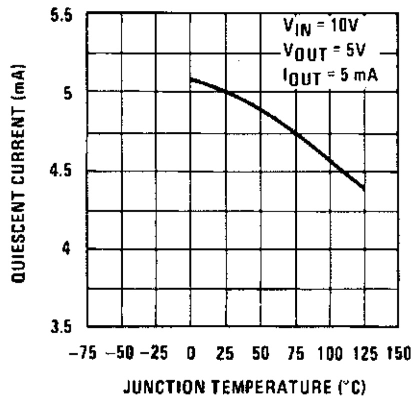
DS007746-12

Dropout Characteristics



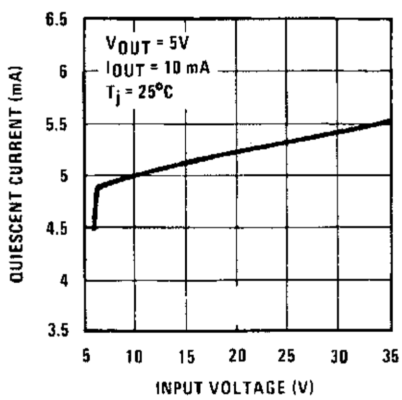
DS007746-13

Quiescent Current



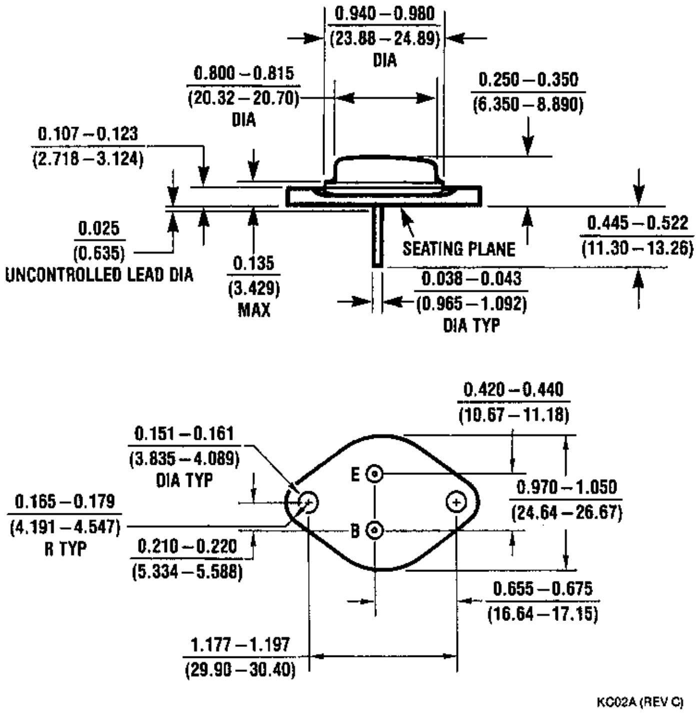
DS007746-14

Quiescent Current



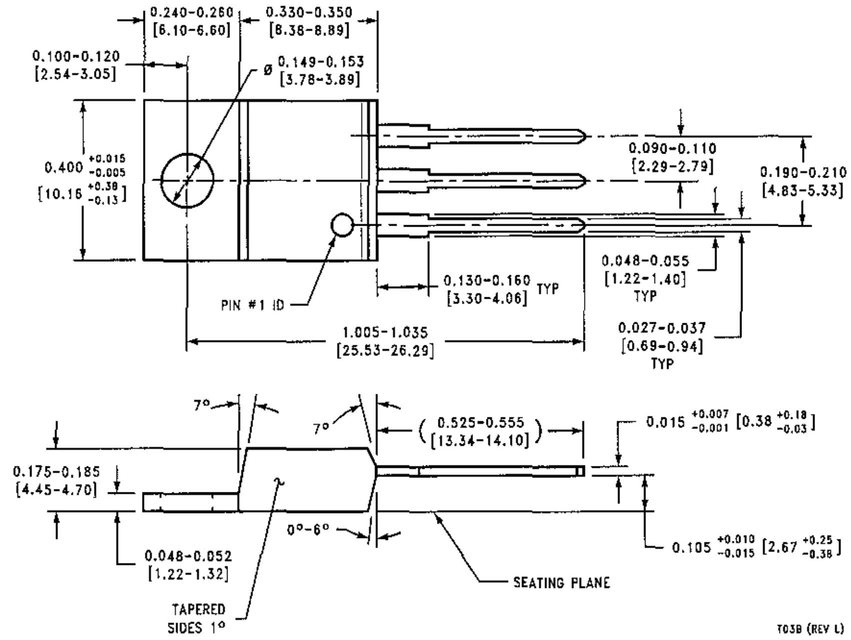
DS007746-15

ıysical Dimensions inches (millimeters) unless otherwise noted



Aluminum Metal Can Package (KC)
Order Number LM7805CK, LM7812CK or LM7815CK
NS Package Number KC02A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



TO-220 Package (T)
Order Number LM7805CT, LM7812CT or LM7815CT
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

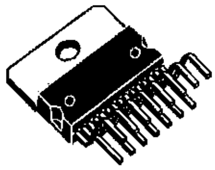
National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

APPENDIX E

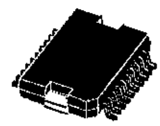
L298N H-BRIDGE DRIVER CHIP DATASHEET

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



Multiwatt15



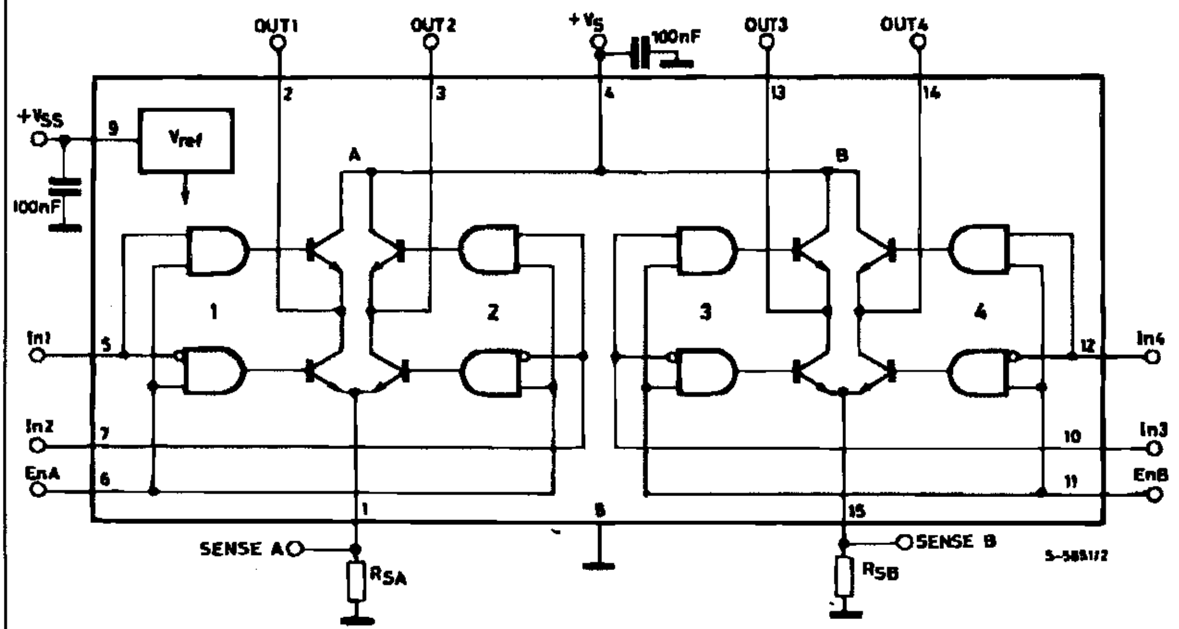
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

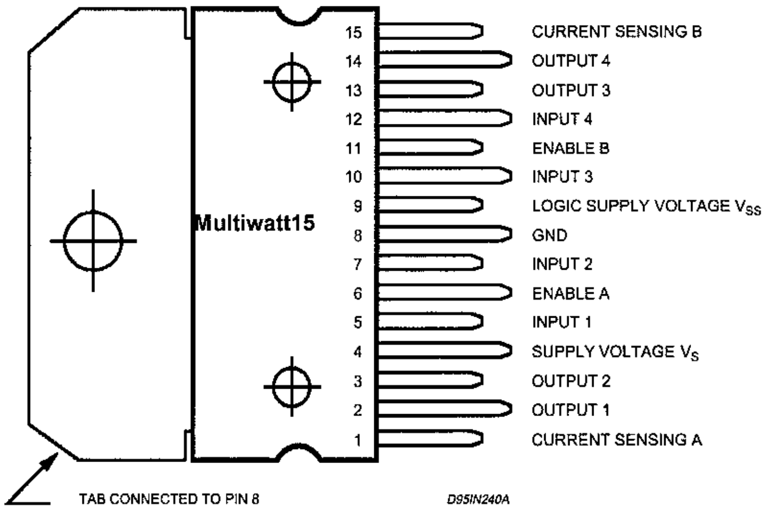
BLOCK DIAGRAM

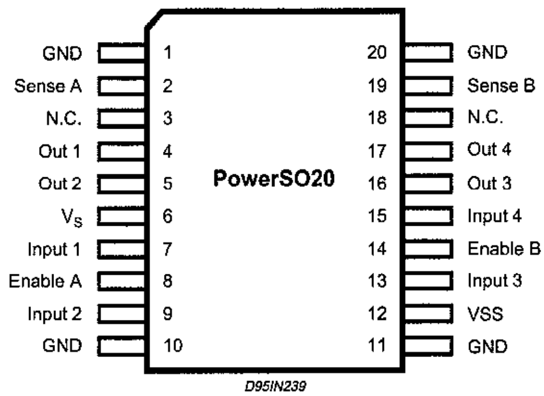


ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100μs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L			–10	μA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V		30	100	μA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	μA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	μA
$V_{CEsat(H)}$	Source Saturation Voltage	I_L = 1A I_L = 2A	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat(L)}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		μs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		μs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		μs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		μs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		μs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		μs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		μs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		μs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		μs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		μs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		μs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		μs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		μs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		μs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		μs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		μs

1) 1)Sensing voltage can be –1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ –0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

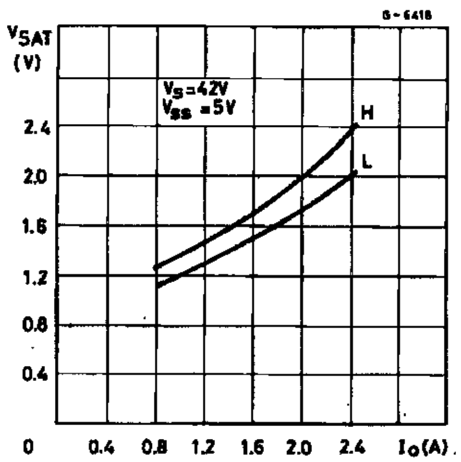
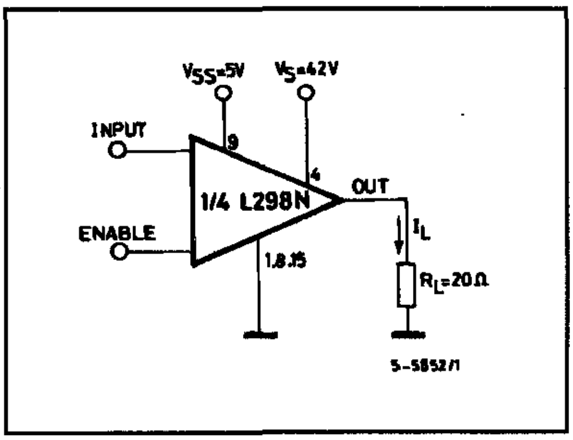


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

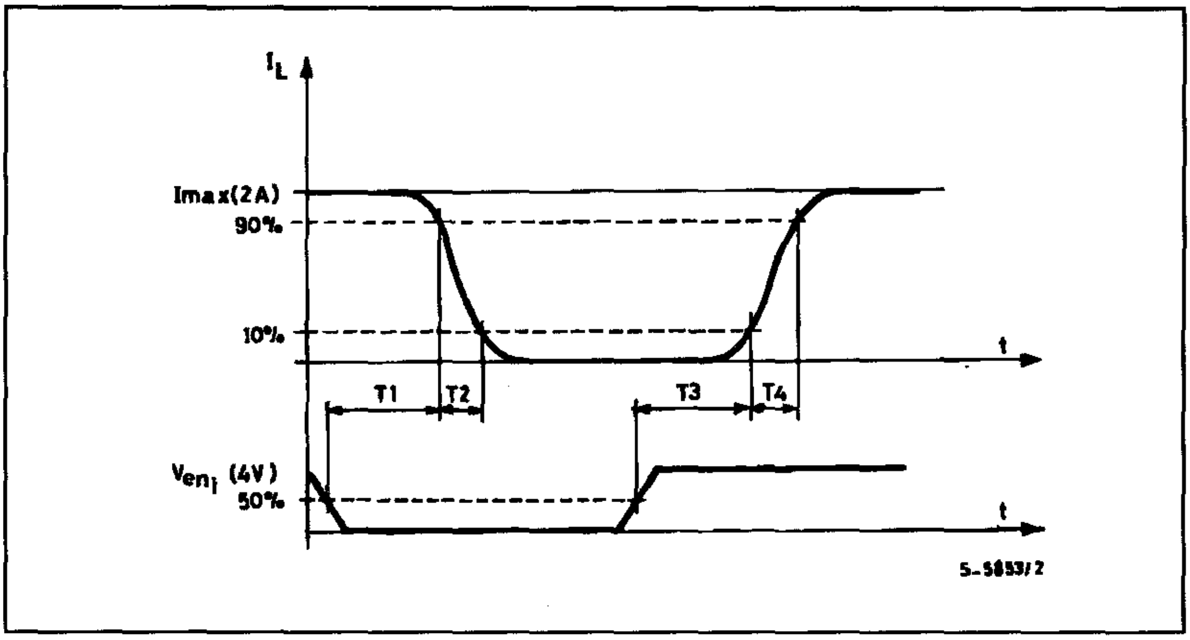
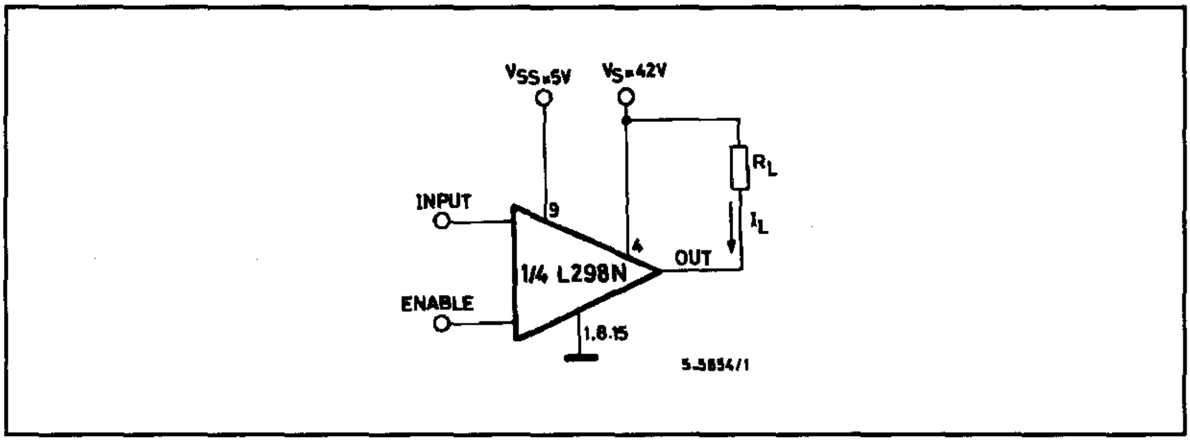


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

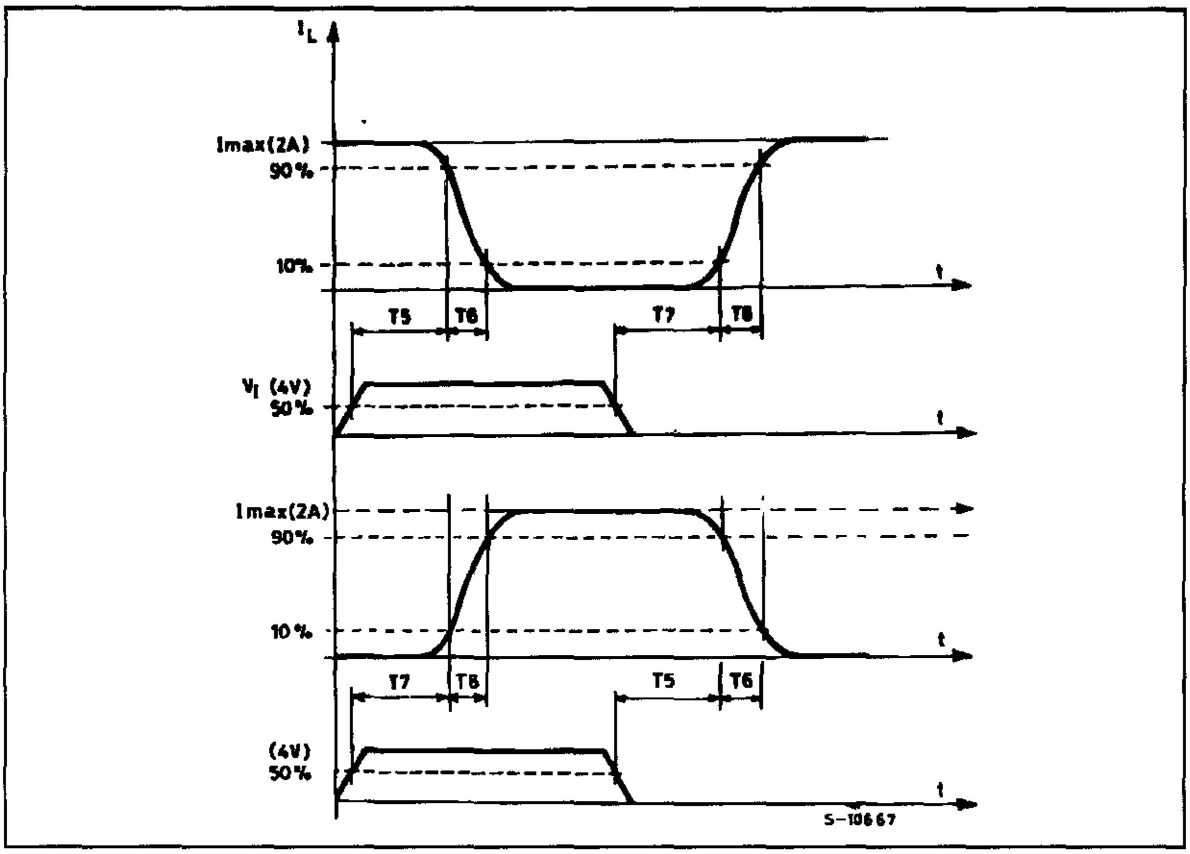
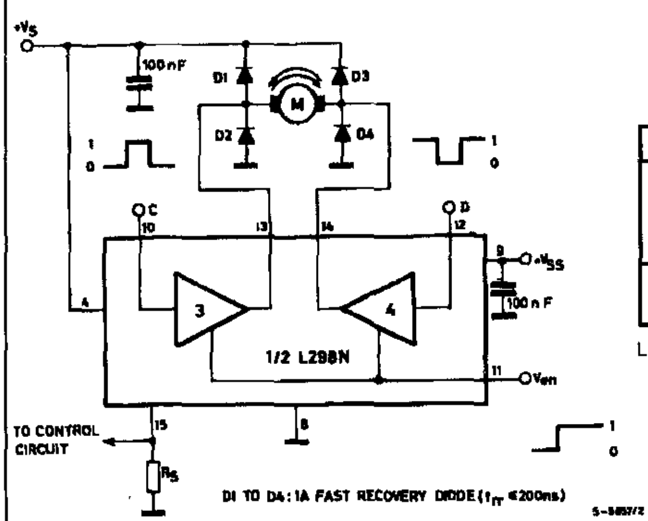


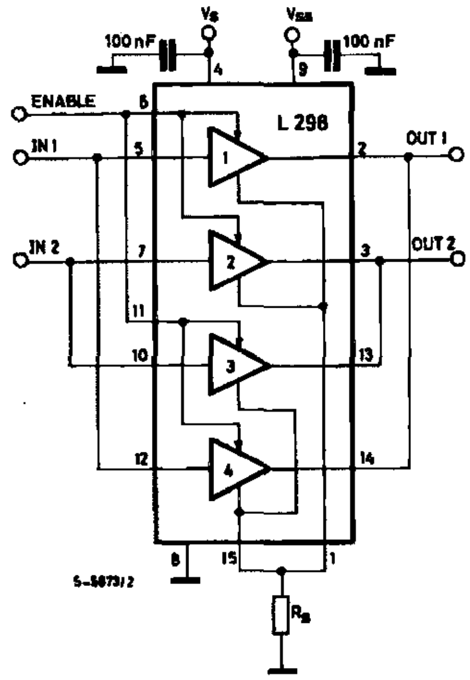
Figure 6 : Bidirectional DC Motor Control.



Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}.) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Shottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

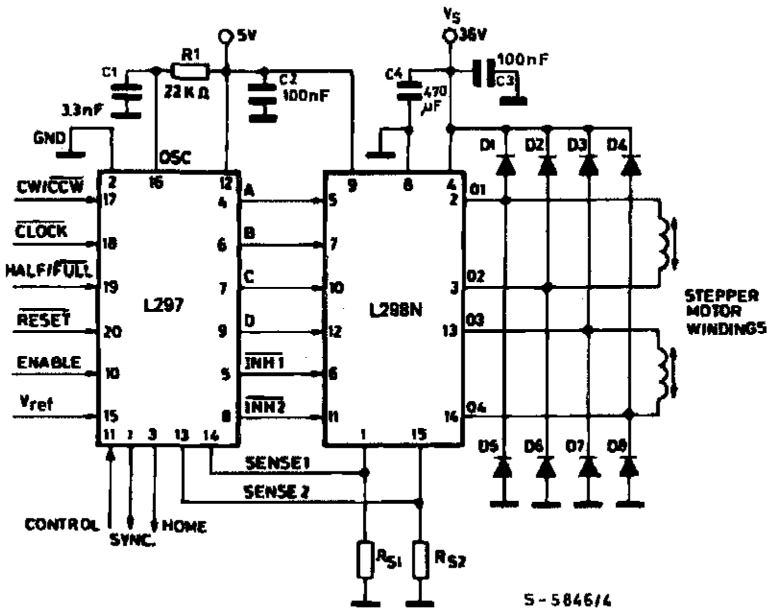
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



$R_{S1} = R_{S2} = 0.5\,\Omega$

D1 to D8 = 2 A Fast diodes { $V_F \leq 1.2$ V @ I = 2 A; trr $\leq$ 200 ns }

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

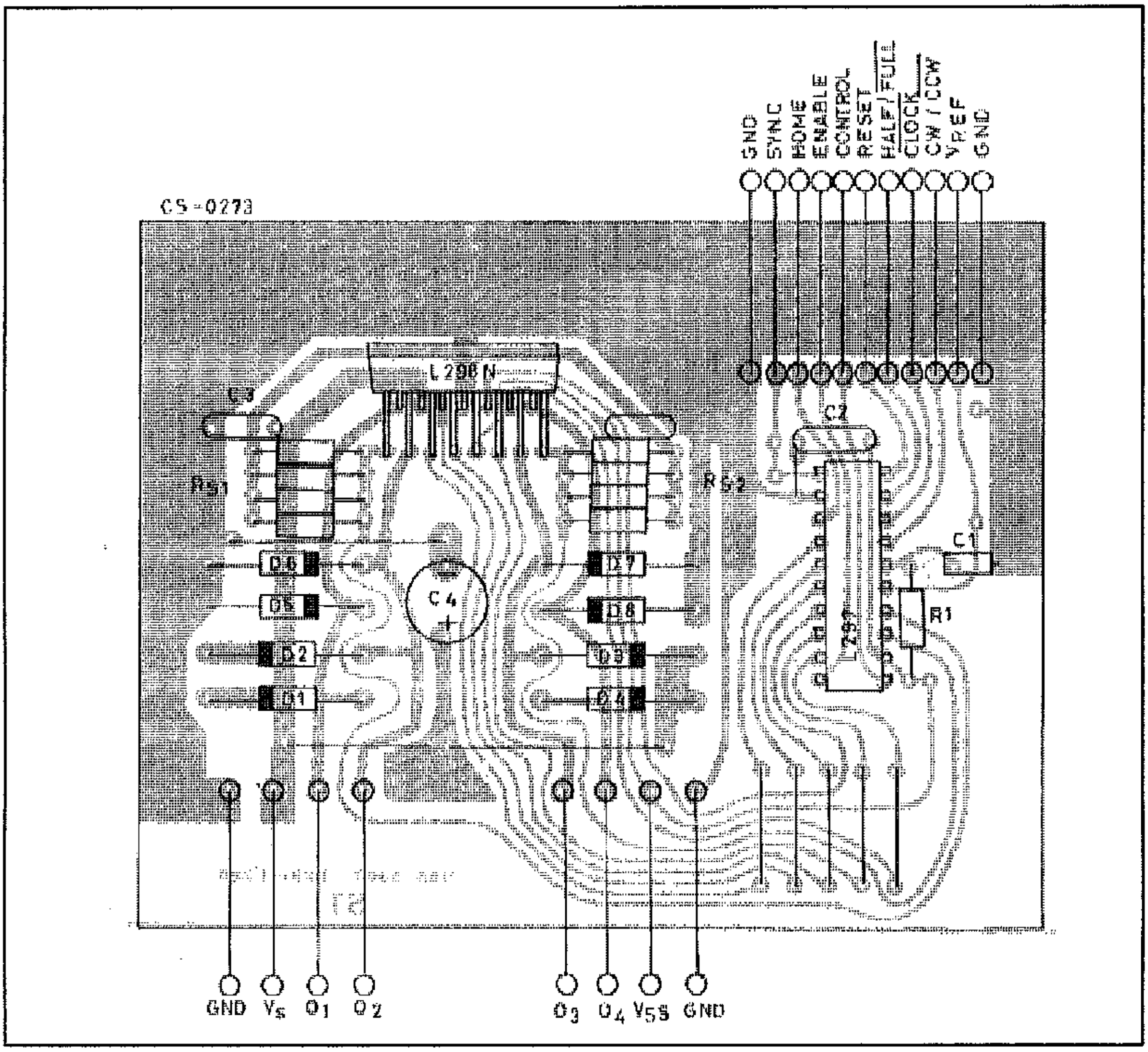
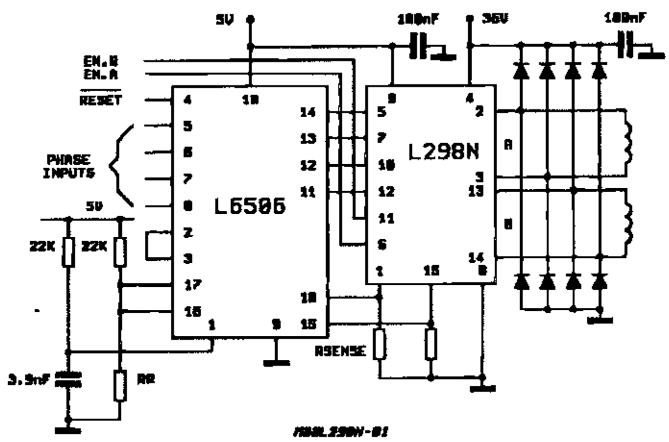


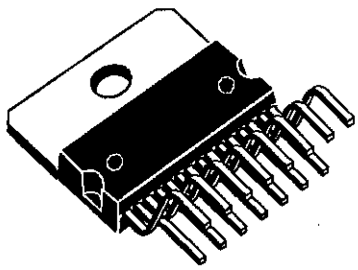
Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



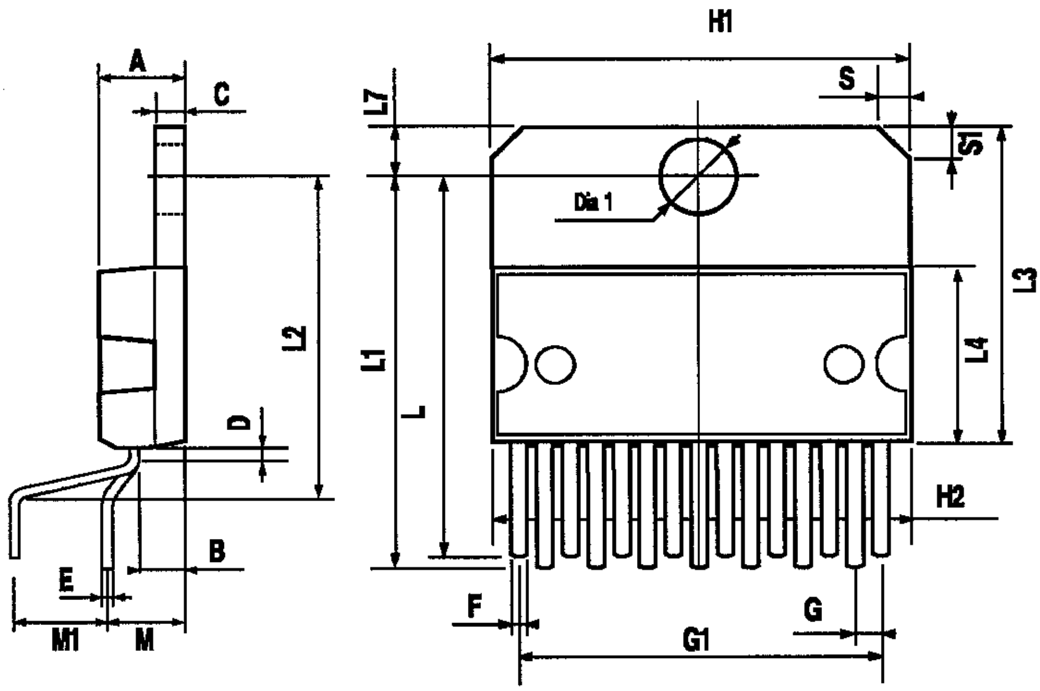
R_R and R_{sense} depend from the load current

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

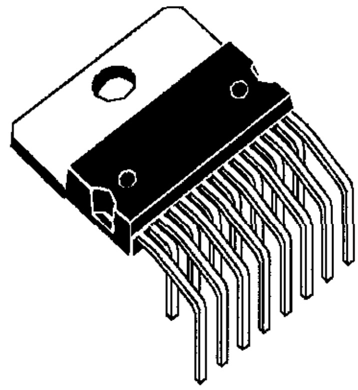


Multiwatt15 V

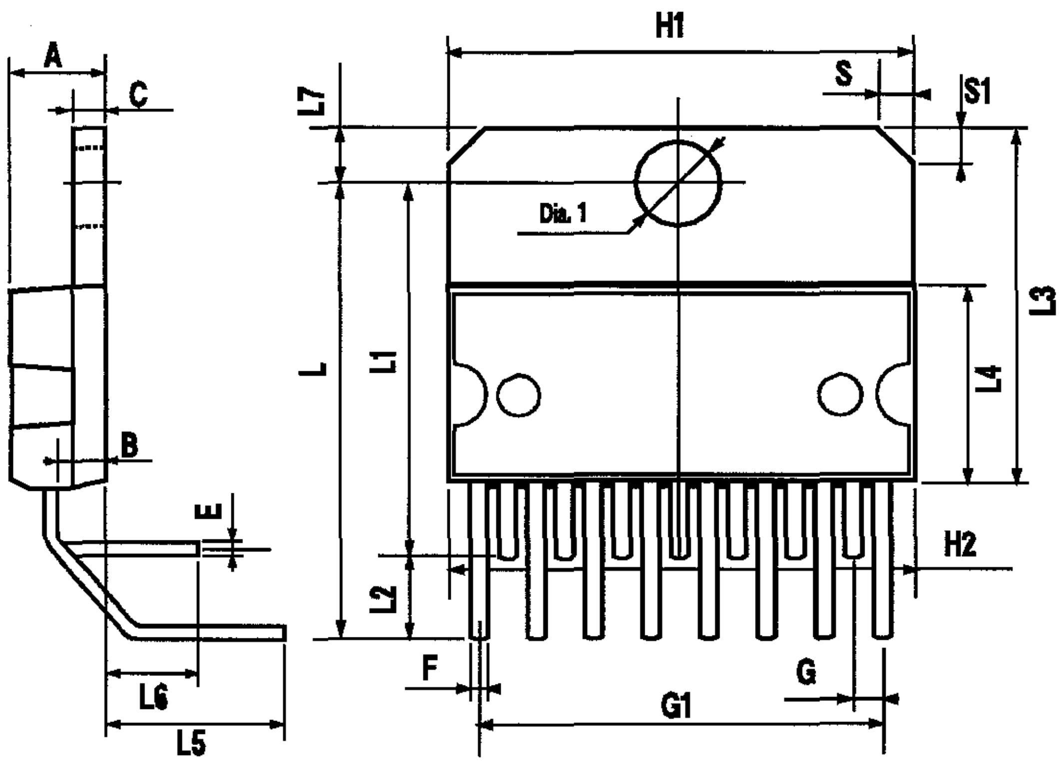


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



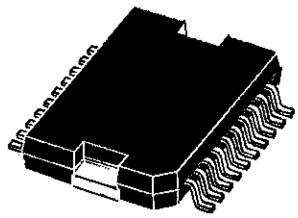
Multiwatt15 H



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

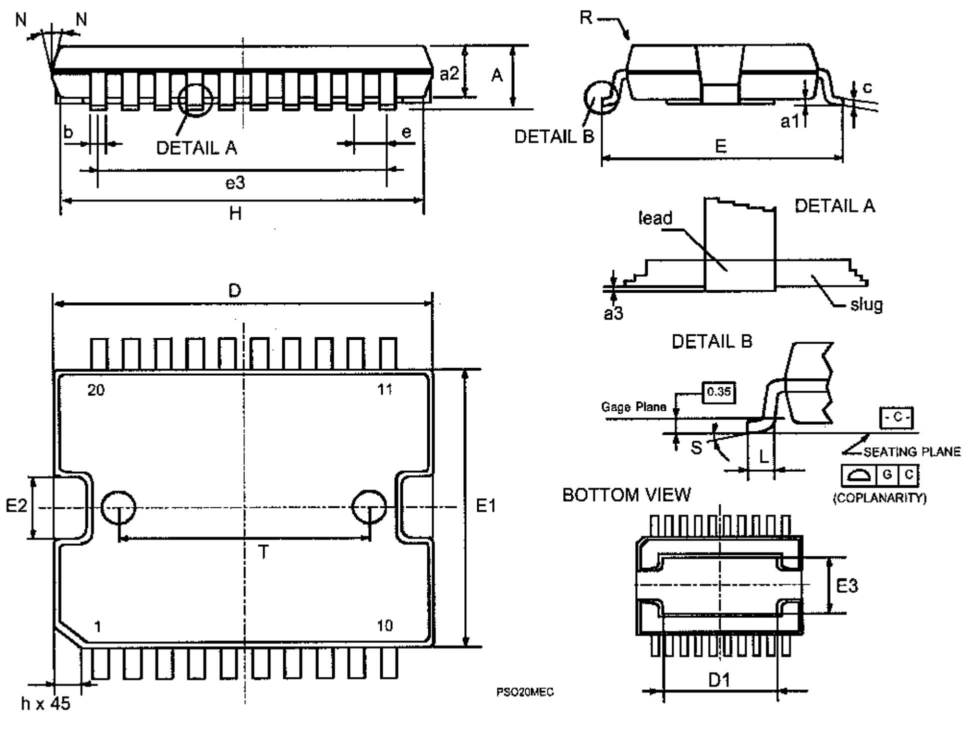
(1) "D and F" do not include mold flash or protrusions.
- Mold flash or protrusions shall not exceed 0.15 mm (0.006").
- Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20



STMicroelectronics GROUP OF COMPANIES

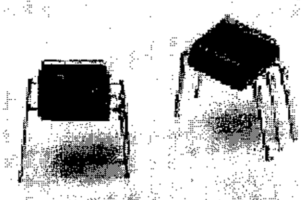
Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

http://www.st.com

APPENDIX F
PHOTOREFLECTOR SENSORS DATASHEET

hotoreflector
5587, P5588

Photo IC output (digital) photoreflectors



587 and P5588 are photoreflectors combining a high power infrared LED and low voltage photo IC. The photo IC consists of a high sensitivity otodiode, amplifier, schmitt trigger circuit, and output phototransistor, etc. on a single chip.

Features

- Miniature package
- Low voltage operation
- Photo IC, open collector output
- P5587: "H" level output at light input
  P5588: "L" level output at light input

Applications

- Paper detection in copiers and printers, etc.
- Tape end detection in VTRs, tape recorders, etc.

■ Absolute maximum ratings (Ta=25 °C)

Parameter		Symbol	Value	Unit
Input (LED)	Forward current	I_F	50	mA
	Reverse voltage	V_R Max.	5	V
	Power dissipation	P	80	mW
Output (photo IC)	Supply voltage	Vcc	-0.5 to +7	V
	Output voltage	Vo	-0.5 to +7	V
	Output current	Io	8	mA
	Power dissipation	P	80	mW
Operating temperature		Topr	-25 to +85	°C
Storage temperature		Tstg	-30 to +85	°C
Soldering		-	260 °C, 3 s, refer to Dimensional outline	-

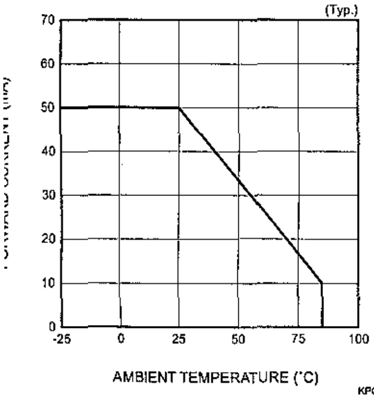
■ Electrical and optical characteristics (Ta=25 °C, Vcc=5 V, unless otherwise noted)

Parameter		Symbol	Condition	P5587 Min.	P5587 Typ.	P5587 Max.	P5588 Min.	P5588 Typ.	P5588 Max.	Unit
Input (LED)	Forward voltage	V_F	I_F=20 mA	-	1.23	1.45	-	1.23	1.45	V
	Reverse current	I_R	V_R=5 V	-	-	10	-	-	10	μA
	Terminal capacitance	Ct	V=0 V, f=1 MHz	-	30	-	-	30	-	pF
Output (photo IC)	Supply voltage	Vcc		2.2	-	7	2.2	-	7	V
	Low level output voltage	V_{OL}	I_{OL}=4 mA *1	-	0.1	0.4	-	0.1	0.4	V
	High level output current	I_{OH}	Vo=5 V *2	-	-	10	-	-	10	μA
	Current consumption	Icc		-	1.3	3.0	-	1.3	3.0	mA
Transfer characteristics	L→H Threshold input current	I_{FLH}	R_L=1.2 kΩ, d=3 mm Reflecting surface: white paper (reflectivity 90 % or more)	-	-	10	-	-	-	mA
	H→L Threshold input current	I_{FHL}		-	-	-	-	-	10	mA
	Hysterisis	-	*3	-	0.8	-	-	0.8	-	-
	L→H Propagation delay time	t_{PLH}	I_F=15 mA R_L=1.2 kΩ d=3 mm	-	-	20	-	-	30	μs
	H→L Propagation delay time	t_{PHL}		-	-	30	-	-	20	μs
	Rise time	tr		-	0.07	-	-	0.07	-	μs
	Fall time	tf		-	0.03	-	-	0.03	-	μs

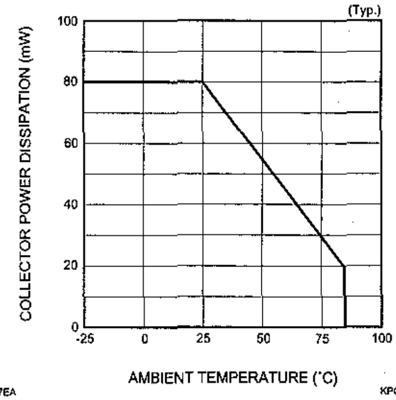
*1: P5587: I_F=0 mA, P5588: I_F=15 mA
*2: P5587: I_F=15 mA, P5588: I_F=0 mA
*3: P5587: I_{FHL}/I_{FLH}, P5588: I_{FLH}/I_{FHL}

Note) Connect a 0.01 μF capacitor or larger between Vcc and GND.

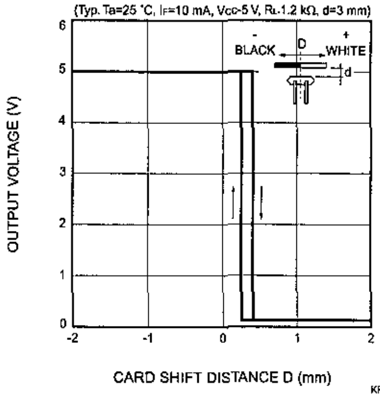
LED forward current vs. ambient temperature



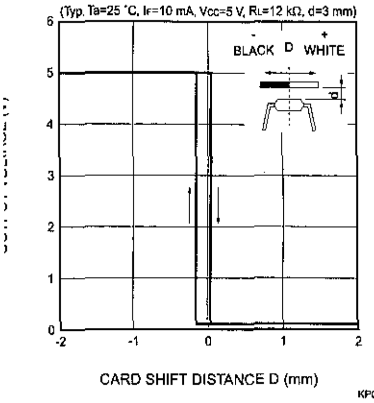
■ Photo IC power dissipation vs. ambient temperature



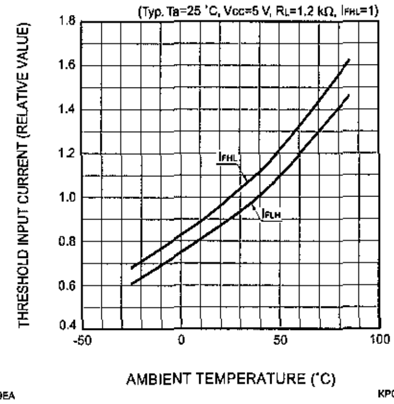
■ Position detection characteristic (P5587)



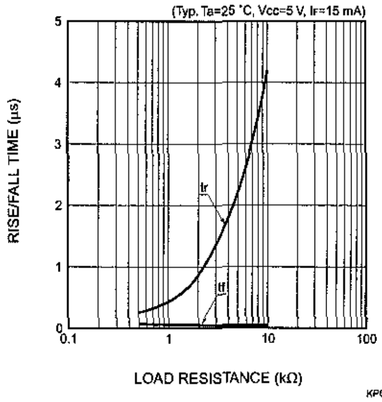
Position detection characteristic (P5587)



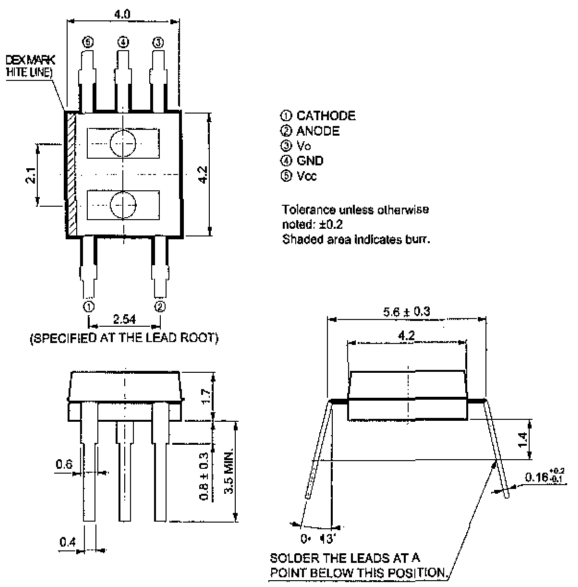
■ Threshold input current vs. ambient temperature (P5587)



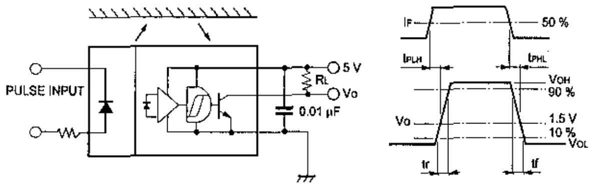
■ Rise/fall time vs. load resistance (P5587)



Dimensional outline (unit: mm)



■ Response time measurement circuit (P5587)



TSU PHOTONICS K.K., Solid State Division
iino-cho, Hamamatsu City, 435-8558 Japan, Telephone: (81) 053-434-3311, Fax: (81) 053-434-5184, http://www.hamamatsu.com
ımatsu Corporation: 360 Foothill Road, P.O.Box 6910, Bridgewater, N.J. 08807-0910, U.S.A., Telephone: (1) 908-231-0960, Fax: (1) 908-231-1218
ımamatsu Photonics Deutschland GmbH: Arzbergerstr. 10, D-82211 Herrsching am Ammersee, Germany, Telephone: (49) 08152-3750, Fax: (49) 08152-2658
amatsu Photonics France S.A.R.L.: 8, Rue du Saule Trapu, Parc du Moulin de Massy, 91882 Massy Cedex, France, Telephone: 33-(1) 69 53 71 00, Fax: 33-(1) 69 53 71 10
om: Hamamatsu Photonics UK Limited: 2 Howard Court, 10 Tewin Road, Welwyn Garden City, Hertfordshire AL7 1BW, United Kingdom, Telephone: (44) 1707-294888, Fax: (44) 1707-325777
: Hamamatsu Photonics Norden AB: Smidesvägen 12, SE-171 41 Solna, Sweden, Telephone: (46) 8-509-031-00, Fax: (46) 8-509-031-01
atsu Photonics Italia S.R.L.: Strada della Moia, 1/E, 20020 Arese, (Milano), Italy, Telephone: (39) 02-935-81-733, Fax: (39) 02-935-81-741

Cat. No. KPC1003E02
May 2001 DN