MOBILE STEREO-VISION OBJECT TRACKING ROBOT

By

MUNIRA NUR FARIZAN BT ISHAK

FINAL DISSERTATION

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

STEREO-VISION OBJECT TRACKING ROBOT

by

Munira Nur Farizan Bt Ishak

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

Approved:

Mr. Patrick Sebastian
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

December 2007

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.

Munira Nur Farizan Bt Ishak

ABSTRACT

Building a mobile robot with sight capabilities enables the robot to do simple tasks of following a straight line or avoiding an obstacle, which in this project, a more complicated task of following an object or human. This project is carried out exploring the sight capability of the robot through computer vision, which implements image and video processing, and also the stereo-vision concept which gives depth perception to the sight of the robot. The primary objective is to develop a mobile robot that has the capabilities to do object tracking in real time, with stereo-vision, utilizing two separate cameras. The movement of the robot is programmed using C programming for the microcontroller, which controls DC motors built with a gearing system on the mobile robot. This program enables robot to move in left, right and forward directions. The object tracking and image processing is developed using the MATLAB program. Image acquisition is done using web cameras. A reference image is saved, from which the object is tracked. Once object is tracked, MATLAB sends instructions to the microcontroller via serial communication, RS232. Different techniques of digital image processing are used such as edge detection and correlation. The project methodology stresses on the research and analysis, circuit design, circuit modeling, microcontroller programming, circuit fabrication and finally, designing the interfacing software. The end system comprises of the hardware for the mobile robot and interfacing software for object tracking.

ACKNOWLEDGEMENTS

I would like to express my gratitude to all those who gave me the possibility to complete my Final Year Project of the Electrical and Electronics Engineering. I would like to thank Universiti Teknologi PETRONAS and the Final Year Project Committee Members for giving me this opportunity.

I am deeply indebted to my immediate supervisor and lecturer, Mr. Patrick Sebastian, from the Electrical and Electronics Engineering (EE) department, and also his colleagues, whose help, suggestions and encouragement helped me throughout the completion of the project. His professionalism and continuous guidance in all given tasks are very much appreciated.

Special thanks to all lecturers and technicians in the EE department, especially Siti Hawa, for their dedication and willingness to provide useful pointers that helps me to understand the flow and technicality of my project.

My student colleagues with similar projects, who supported me with the project, I want to thank them for all their help, support, interest and valuable hints. Working with them has helped me overcome obstacles of the project, through valuable discussions.

Especially, I would like to give my special thanks to my family and friends whose daily encouragement and love enabled me to complete this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Robotics is the branch of technology that deals with the design, construction, operation, and application of robots [17]. In the last decade, the robotics field has encountered a new trend in which new applications are gaining more interest in the commercial area. Mobile robots were exclusive research topics for the academic and military world. But as certain areas in the leisure industry became more and more important in the society the idea of mobile robots are also inspiring commercial companies. [5]

Many applications would benefit from fully autonomous systems (mobile robots). The movements of robots are limited to simple actions but integrated with the ability to see brings the robot to the next level. This is known as machine vision or computer vision, which is the capability to sense, store and reconstruct a graphic image that matches the original as closely as possible [11]. The craze of making robots more humanized has inspired many to develop robots that have the senses of a human such as seeing, hearing and touching.

The sight ability is brought for further developments with the help of the computer visions technology with stereo vision that enables 3D vision on the robot. Over the past decade, the geometry of multiple views, (focusing on two-views in this project), in computer vision has developed rapidly, where results have been achieved when processing tasks and algorithm using two images simultaneously, to create a 3D image [11].

1.2 Problem Statement

Building a mobile robot with sight capabilities enables the robot to do simple tasks such as following a straight line or avoiding an obstacle. These capabilities seem very limited when considering the robot of having sight capabilities. The ability of the robot of being able to 'see' moving objects, such as a human, especially in real time need brings the robot to a new level. Projects in security and surveillance have been worked on in developing the capability of tracking a moving object, or most importantly a human. The computer vision technology can be implemented in the design of the mobile robots to enhance their sight capabilities by programming the robot to be able to move according to the object tracked. The robot must also have the flexibility of being able to move swiftly in any direction of the moving object.

Stereo vision has been widely researched and used in the industries. Stereo vision mimics the capabilities of the human vision enables robot to get depth information. The sight and object tracking of the robot can be further enhanced with stereo vision in terms of recognizing the object tracked with depth perception, thus enabling the robot to actually recognize the object that it is 'seeing'. Implementation of the stereo vision in real time on the mobile robot would require interfacing between the hardware of the robot and the camera utilized.

The mobility of the robot would require work as to how fast the robot reacts to signals given in real time. As the object tracked would be moving randomly, fast reactions of the robot is required.

1.3 Problem Solution

This project focuses on computer vision, the microcontroller and robotics studies. The mobile robot designed has the capability of stereo vision object/person tracking utilizing two separate webcams as the image acquisition device. The mobility of the robot is controlled by a microchip that controls motors to move the wheels of the robot. The robot is able to move in four directions, front, back, left and right. The operations of the robot is developed using C programming on the microcontroller.

The two cameras used captures movements of objects within sight and the movement of the robot is determined by the movements of the object tracked. The image acquisition is processed using the MATLAB software where images acquired is processed on a real time basis. The communications between the robot hardware and the cameras is interfaced with a PC, using serial communication between the cameras and the PC.

1.4 Objectives and Scope of Study

1.4.1 Objectives

This project has the objectives of:

1 Developing a mobile robot that is able to move forward, backwards, left and right with desired speed
2 Developing an object/person tracking algorithm for real time tracking utilizing webcams and image processing techniques
3 Developing a stereo-vision algorithm utilizing two cameras to achieve depth perception of objects tracked
4 Developing an algorithm which enables MATLAB to communicate with the robot

1.4.2 Scope of Study

The scope of this project includes the studies of computer vision, the microcontroller and robotics. Computer vision exploration utilizes the MATLAB software for digital image processing. MATLAB is used to develop algorithms for object tracking and stereo vision from real-time images obtained from the webcams used.

The area of automation and robotics is explored in developing the mechanisms for the mobile robot to be able to move from one location to another smoothly with specified speed when commanded by instructions from the signals of the cameras. These instructions are processed by the mobile robot using the microcontroller to control the two motors that are used in the mechanisms of the movements of the robot.

Communication method that is used is serial communication. Communication is crucial when transferring data from the cameras to the PC to be processed by MATLAB. Instructions are sent back to the robot microcontroller for the movement of the robot tracking the object.

CHAPTER 2

LITERATURE REVIEW AND THEORY

2.1 Object Tracking

Object tracking uses image processing to utilize captured images and be able to track movements during the given time. A simple algorithm for object tracking by a fixed camera compares the current image with a reference image and simply counts the number of different pixels. More complex algorithms are necessary to detect motions when the camera itself is moving, which is the case for this project, or when the motion of a specific object must be detected in a field containing other movements which can be ignored.

2.1.1 Correlation

The basic object tracking is done using phase correlation using MATLAB. This technique involves correlating two images by first performing a two dimensional Fourier Transform on each image. One of the images is the reference image and the second one is the shifted image of the first image. These images are obtained from a breakdown of images from a video. After performing the Fourier Transform, the image data transforms into frequency domain. The first image data is multiplied by the conjugate of the second image data and these images are then normalized. Inverse Fourier Transform is performed to get phase differences. The final result would be a correlation surface that will have a peak at the coordinates corresponding to the shift of the object. [9]

Cross correlation is another method of object tracking, correlating two images in spatial domain. The method to go forward with in this project is further researched and experimented with to determine best methods to be used.

2.1.2 *Edge Detection*

A technique of image segmentation called edge detection can be used in this project as a form of recognizing target objects to be tracked. In the previous semester, the target detection program was studied from which a small segment of an image is used to detect its position from its original image. This program uses the function *edge* to achieve this. [4]

2.1.3 *Object Recognition in Videos*

A method of object recognition in videos is researched where an object can be tracked in the video regardless of the position or angle. This uses the method of 'viewpoint invariant descriptors' to recognize the object where several of viewpoints are recorded for recognition of object. Later the regions are tracked to reject unstable region and to reduce noise, this uses the text retrieval method. Research is done on 3D object recognition and text retrieval methods to be able to apply this method on a video. This method is called Video Google and readings are found in [10]

2.1.4 *VCAPG2*

The function VCAPG2 on MATLAB enables interfacing with multiple webcams. VCAPG2 can be used with multiple cameras. Interface for the webcam with MATLAB is then further developed using vcapg2 for multiple camera usage as stereo vision needs to be implemented.

2.2 Stereo Vision

Research is done on stereo vision and how it affects object tracking. Stereo vision is a method used in mobile robotics to detect obstacles. Objects appear to be 3D with stereo images. Stereo cameras are used to get depth information which enables robot to achieve interaction in the 3D space.

Stereo vision is used for mobile robots as it is a reliable and effective way to extract range information from the environment (real-time implementation on low-cost hardware).

2.3 Mobile Robot

There are many different designs of mobile robots that can be implemented in this project. Amongst the many designs, examples of two of the most common designs of robots that can be implemented are:

- Legged robots; very popular in mimicking human or animal movements
  - Bipedal
  - Tripod
  - Quadrapod
  - Hexapod
- Wheeled robot; very convenient in achieving speed and mobility
  - Balancing robot (2 wheeled)
  - 3-wheeled
  - All terrain (buggy car)

2.3.1 *The Robot Design Matrix*

In order to determine the best robot design, the design matrix are done to evaluate the feasibility of the robot according to the relevance if the project. Below are criteria that need to be considered when considering the robot design:

1. **Mobility**
   a. Rate 10 for most mobile, 1 for not mobile
   b. Robot is able to move freely forward, left and right to follow object/person tracked
2. **Speed**
   a. Rate 10 for high speed, 1 for low speed

b. Robot is able to move in reasonable high speed

3. **Cost**
    a. Rate 10 for low cost, 1 for high cost
    b. Low cost but high quality robot
4. **Portability**
    a. Rate 10 for most portable, 1 for least portable
    b. Robot can be transported around easily
5. **Size**
    a. Rate 10 for reasonable (smaller size), 1 for unreasonable size (larger size)
    b. Size of robot must be suitable as a tracking robot and not too big.
6. **Uniqueness**
    a. Rate 10 for most unique, 1 for least unique
    b. A design that is unique from previous robot designs done
7. **Appearance**
    a. Rate 10 for most appealing, 1 for least appealing
    b. Design is appealing to users
8. **Complexity**
    a. Rate 10 for least complex, 1 for most complex
    b. Design must not be too complex to build within the given time frame

Table 1 is created to weigh out the criteria against all available designs; at least four designs. Once the scores are given and have been totaled up, the one with the highest score is the chosen design. However, should the winning design appears not be irrelevant to the project, some adjustments need to be made to the criteria and weights in the given scoring.

Table 1 Design Matrix Table

Criteria	Design 1 (Bipedal)	Design 2 (Tripod)	Design 3 (Hexapod)	Design 4 (3-wheeled)	Design 5 (3-wheeled/legged)
Mobility	5	3	2	10	10
Speed	4	3	6	10	10
Cost	1	6	7	7	2
Portability	6	5	7	8	5
Size	6	5	8	8	5
Uniqueness	9	8	7	3	9
Appearance	9	9	6	7	7
Complexity	2	2	4	10	1
Weight Total(x/80)	**42**	**42**	**47**	**63**	**49**

2.3.2 Robot Design

2.3.2.1 Chassis/base

5-8 inch diameter base for better mobility.

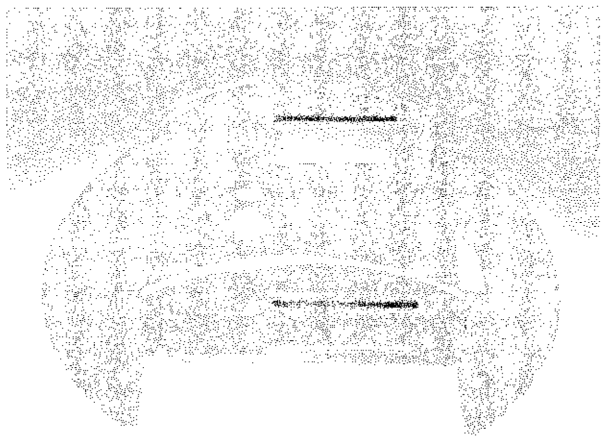


Figure 1 Mobile Robot Base

2.3.2.2 Castor wheel

Castor wheel to balance the base

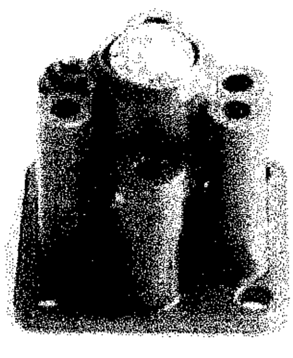


Figure 2 Castor Wheel

2.3.2.3 *Wheels*

- A pair of Tamiya wheels with 50mm diameter and 30mm width is used

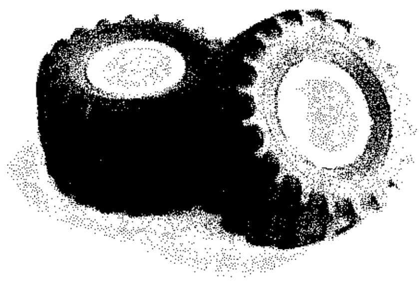


Figure 3 Tamiya Off-Road Wheels

2.4 Circuit and Mechanism Design

2.4.1 *Twin gearbox*

- small (3in long) plastic gearbox
- contains two small DC motors that drive separate 3mm hexagonal output shafts
- two ways to put the kit together: with a high-speed 58:1 gear ratio or with a slower 203:1 gear ratio
- provide plenty of power to drive a small robot

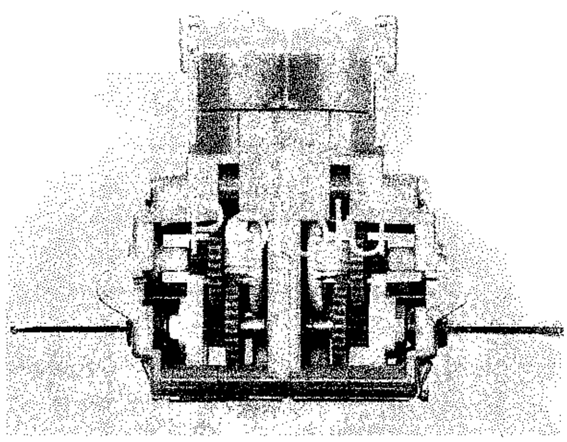


Figure 4 Tamiya Twin Gearbox

2.4.2 *H-Bridge L298N*

Using the H-bridge IC, the initial design of the movement of the robot can be determined. The direction of the motor can be changed and this enables the robot to go in four directions, forward, backwards, left and right.

Below is a diagram of the H-bridge IC connections:

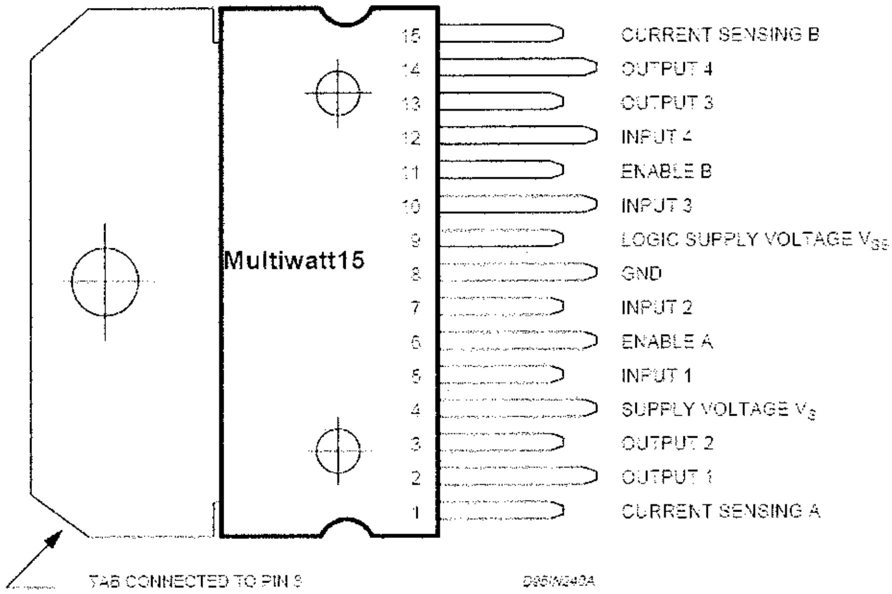


Figure 5 H-bridge IC (L298N)

Please refer to Appendix F for datasheet.

- Input 1-4: input is connected with switches to determine the direction of the motor, 1&2 for motor 1, 3&4 for motor 2.
- Output 1-4: connected to motor, 1&2 for motor 1, 3&4 for motor 2.
- Logic voltage supply Vss : 5V
- Supply voltage Vs: according to motor, usually 12V

- Enable A&B: 1 for enable 0 for disable, this can also be supplied from the PIC for a programmed algorithm.

2.4.3 Microcontroller PIC16F877

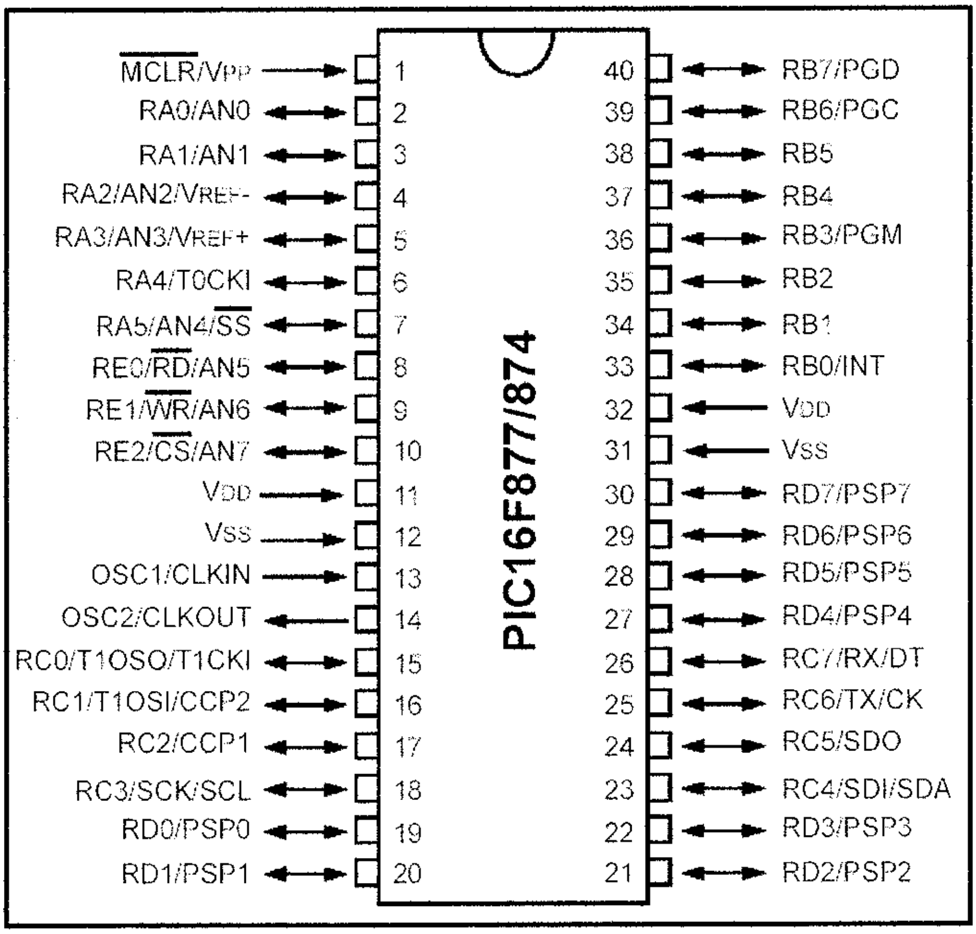


Figure 6 Microcontroller PIC16F877

Please refer to Appendix E for datasheet

- Serial communication is used for control on this circuit to the PC
- MAX 232 is used between the PIC and 9-pin connector

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

Below is the flowchart of the methodology of the project:

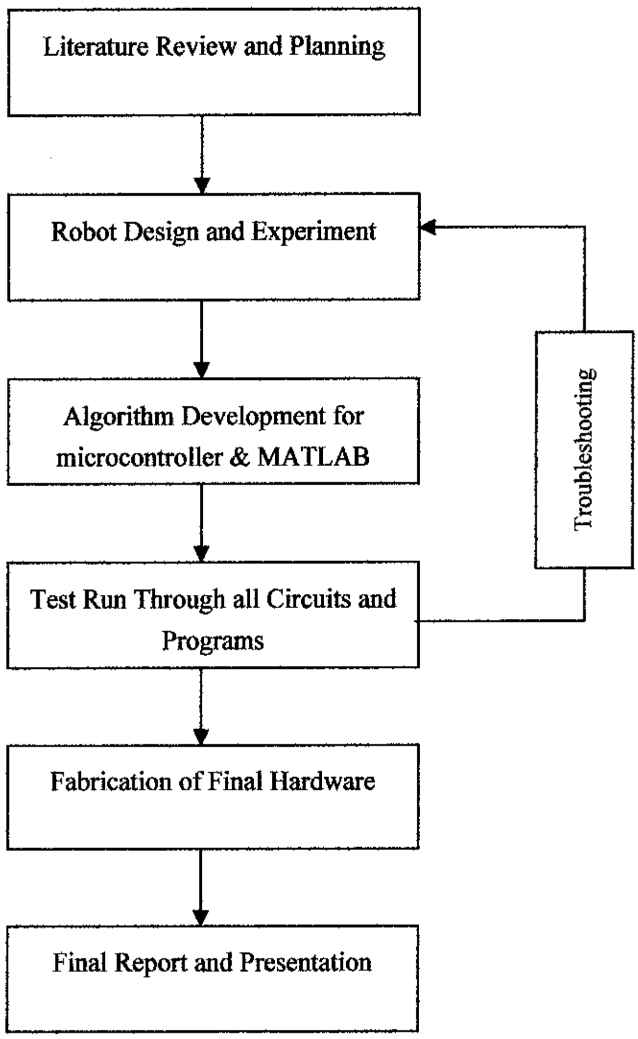


Figure 7 Methodology of Project

3.1.1 *Literature Review and Planning*

The literature review is done to get the initial idea of the project and to assist in the planning and development of the overall project. The scope of the literature review is the research done on object tracking, stereo vision technology and the mobile robot designs. The methods and tools used for the object tracking and stereo vision are hereby discovered through research and supervision by the supervisor before getting into the technical work. Raw materials and parts are surveyed during the planning stage of building of the robot.

3.1.2 *Robot Design and Experiment*

Design is to be done with the support of research and data collected. Mechanisms of the movement of the robot are determined. Previous experiences of utilizing the microcontroller help with the circuit designs. Once the circuit designs are complete, it is implemented and experimented integrated with motors for the robot. Troubleshooting is done after the main testing of the circuit is done.

3.1.3 *Algorithm Development for Microcontroller & MATLAB*

The algorithm of the microcontroller and also the algorithm of MATLAB for the object tracking need to be done parallel with the robot and circuit designs to ensure ample time for troubleshooting and modification of the programs later on. The image acquisition and processing algorithm is developed using MATLAB. The algorithm for the movement of robot, interconnected with the MATLAB software is developed for the microcontroller.

3.1.4 *Test Run through all Circuits and Programs*

Once all circuitry and programs are completed, a run through test is done to check on the functionality of the hardware and software. Many steps are taken on this to test different parts separately.

3.1.5 Fabrication of Final Hardware

Once the testing and troubleshooting are completed, the fabrication of the final hardware is done. The functionality of the circuits and programs need to be smooth before proceeding to the final stage of making the prototype presentable.

3.1.6 Final Report and Presentation

The final report is completed and submitted as scheduled and the oral presentation with the appropriate literature research is carried out according to the time scheduled by the board.

3.2 Tools Required

The tools required for this project can be categorized into two, hardware and software which are listed in the next section. The software components involved are mainly for programming and creating algorithm purposes. The hardware components are for the main structure of the mobile robot.

Table 2 Tools Required

No	Hardware	No	Software
1	1 Motor Controller **L298N**	1	MATLAB 6.5.1
2	1 Microcontroller **PIC16F877**	2	PIC C Compiler
3	2 DC motors 1.5V-4.5V rating	3	Eagle Software
4	2 Web cameras 640x480 resolution	4	WARP 13
5	1 RS232 serial communication port		
6	1 PIC Programmer		
7	2 Tamiya wheels 50mm diameter		
8	1 Castor wheel 35mm height		
9	1 Plastic robot base 5-8 inch diameter		
	Other circuitries/components:		
10	1 breadboard		
11	1 4MHz crystal oscillator		
12	1 MAX232		
13	Resistors, capacitors and LEDs		

3.2.1 *Software*

3.2.1.1 MATLAB

The MATLAB software is responsible for the algorithms created mainly for the object tracking part of the project. Some projects from the MATLAB Central website, file exchanges can be downloaded, focusing on image processing [18]. Among the projects downloaded for practice and studying purposes are Video Surveillance Using MATLAB and Image Acquisition Toolbox, Simulink Video Processing B: Motion Detection, Face Detection System, Face Recognition System, Digital Image Correlation and Tracking, Target Detection and Tracking Badminton rackets.

These projects provide the initial ideas for object tracking. For example, should the robot need to detect faces in order to start moving the algorithm on face detection or recognition can be used, and the algorithm for the tracking of badminton rackets can be a guide for a shape recognition algorithm.

3.2.1.2 C Compiler and WARP13

The C Compiler program is used for compiling the algorithms written for the PIC that is used for the robot controls. The program, once written is compiled using this software and programmed into the microcontroller using the WARP13 software and the PIC Programmer.

3.2.2 *Hardware*

3.2.2.1 Circuitries

The main components in the circuit are listed below. Other components are used in this project which is not listed which will be listed later on.

3.2.2.1.1 Motor Controller

The L298N IC is an H-bridge motor controller which enables bi-directional rotation of the motors. For datasheet refer to *Appendix F*

3.2.2.1.2 Programmable Integrated Circuit

Programmable Integrated Circuit (PIC) 16F877 is an enhanced flash memory PIC, with 33 input/output pins, 368 bytes of data RAM and 256 bytes of data EEPROM. For datasheet refer to *Appendix E.*

3.2.2.2 Motor

Two DC motors are used in this project. The motors come with the Tamiya Twin Gearbox set, each with a rating of 1.5-3V. For datasheet refer to *Appendix G*

3.2.2.3 Camera

Webcams with 640 x 480 resolution and frame rate of 30 frames per second are used for stereo vision object tracking purposes.

3.2.2.4 Wheels

A castor wheel and two back wheels attached with motors are used.

3.2.2.5 Outer Shell of Robot

The material used for the outer shell of the robot is Perspex or plastic.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 System Design

4.1.1 Mobile Robot Base

Below is the new base design

Figure 8 Robot Base Design

A test base is used for testing circuitries which is shown in Figure 9:

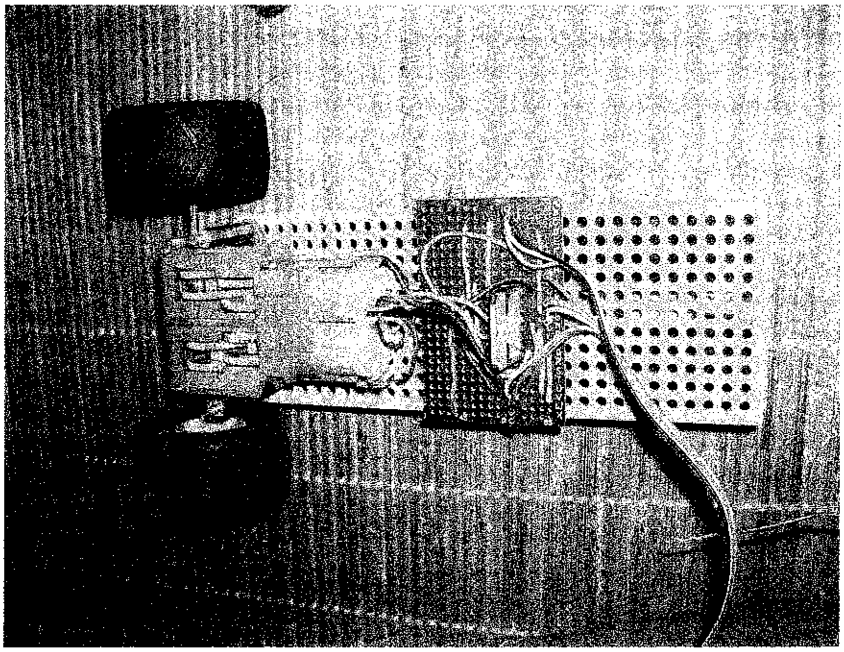


Figure 9 Test Base

4.1.2 Circuit Design

4.1.2.1 H-Bridge

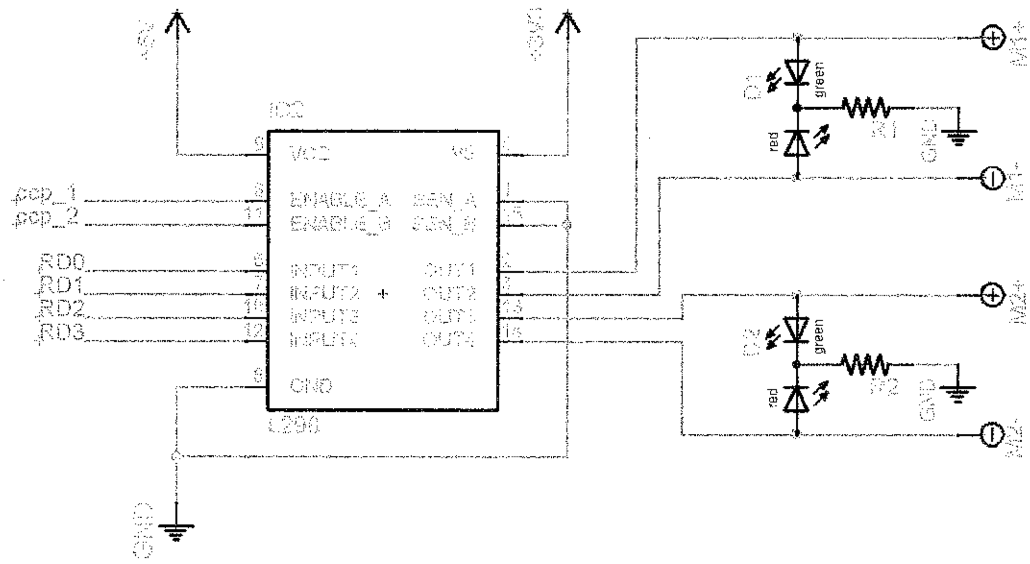


Figure 10 Circuit Diagram for H-Bridge

1 CCP1 and CCP2 are used to control the speed for each motor

2 4 output pins from the PIC16F877 is put as input pins on the L298N

3 Additional circuits with red and green LEDs at the motor is used to determine direction of motor(backward/forward movements)

4.1.2.2 PIC16F877

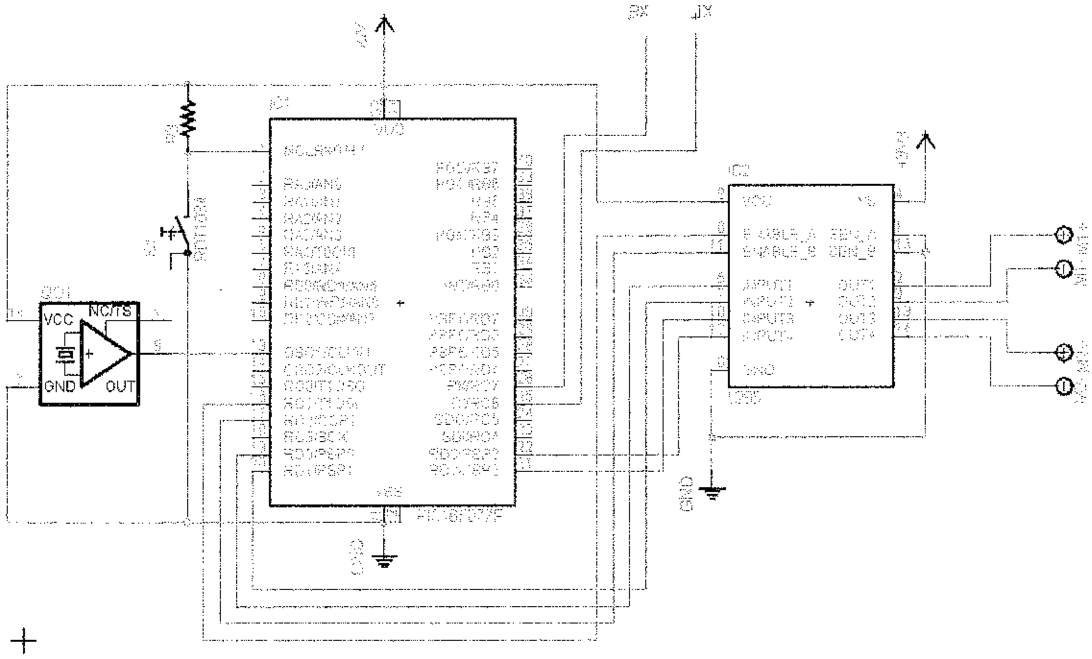


Figure 11 PIC-L298 Circuit Diagram

4.1.2.3 Serial Communication

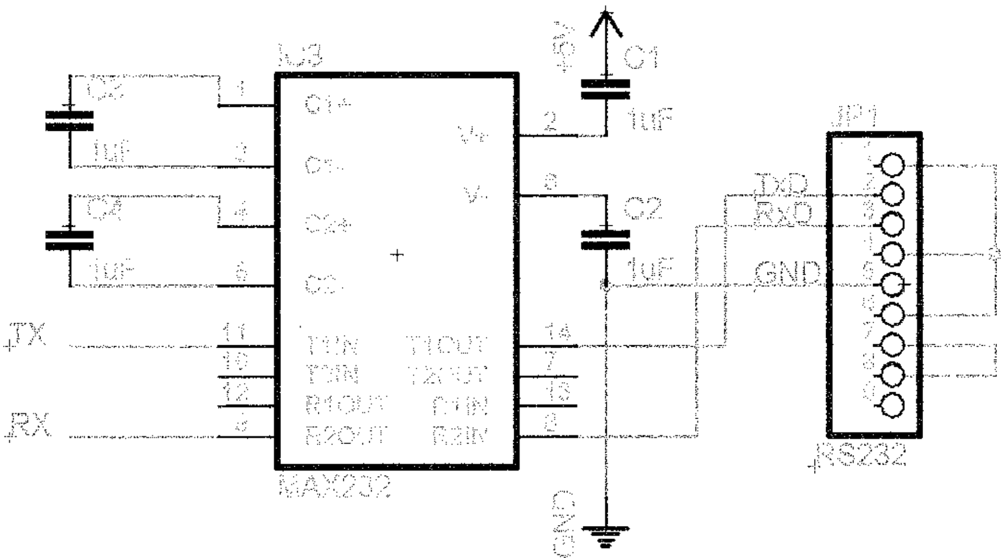


Figure 12 Circuit Diagram for Serial Communication

4.1.3 Robot Movement Algorithm

Using the H-bridge IC, the design of the movement of the robot is determined. The direction of the motor can be changed and this enables the robot to go in four directions, forward, backwards, left and right. Table 3 is the truth table for the H-bridge, taken from the datasheet in *Appendix F.*

Table 3 :Truth table for 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

Here, V_{en} is the enable pin, which is active high. It is set to high to set the motors running and low to disable the motors. From this, the robot movement algorithm is determined.

Table 4 :Robot Movement Algorithm

Input 1	0	0	0	0
Input 2	1	1	1	0
Enable A (Duty Cycle)	100	100	200	0
Input 3	0	0	0	0
Input 4	1	1	1	0
Enable B (Duty Cycle)	100	200	100	0
Robot Direction	Forward	Left	Right	Stop

1 If both motor polarity were the same, direction of both wheels are the same → forward/backward movement

2 Enable A&B input will come from the PIC which will be controlled by the

object tracking algorithm.

3 PWM is used on enable A & B to control the speed, from which the left and right movement can also be controlled.

4.1.3.1 Flow of movement

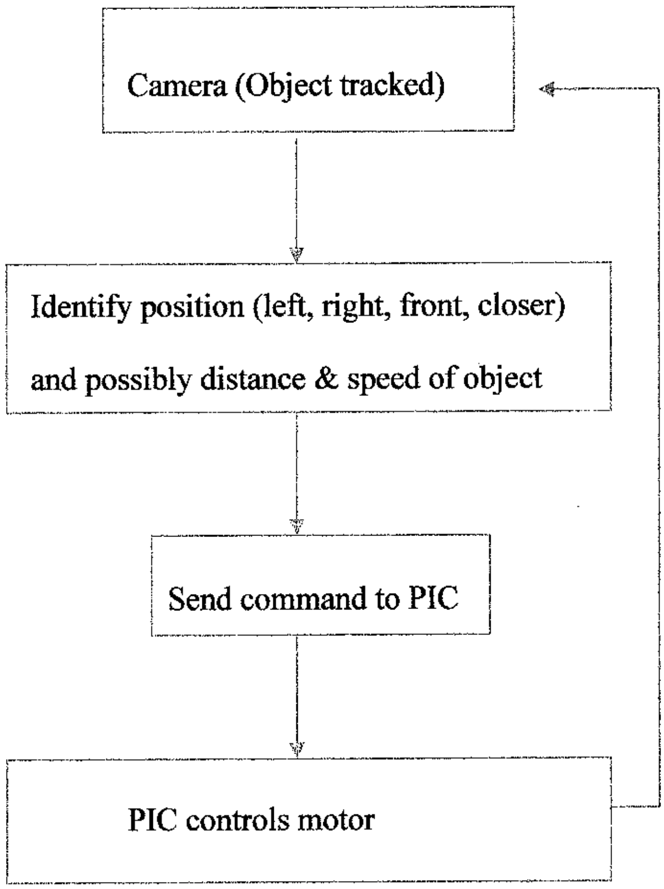


Figure 13 Flow of Mobile Robot Movement

Movement Commands:

(1) Forward → when object is in the middle of camera; moving/not moving

(2) Left → when object moves to the left

(3) Right → when object moves to the right

(4) Speed → speed is used when moving from left to right.
The speed of one motor is higher than the other for left or right movement

4.1.3.2 Motor Controller Program

The program for the motor controller is focused mainly for receiving data from the MATLAB program, via RS232. Only one character length of data is transmitted at one time giving the instruction to the PIC on the direction for the motors to move. In this program, PORT D is used as output.

4.1.3.2.1 Forward movement

When the instruction forward in the form of a character received from the program MATLAB, the PIC program sets the ports on the PIC, RD0 and RD2 to low and RD1 and RD3 to high. This provides the input for the L298, input 1 to 4 and moves both motors in the clockwise direction, thus giving a forward movement. The CCP_1 and CCP_2 are set to a value that will remain constant for the forward movement.

4.1.3.2.2 Left movement

When the instruction left is received from MATLAB, the PIC program sets the ports RD0-RD3 the same as the forward movement. This time, the CCP_2 is set at a higher value than CCP_1. This causes the speed of the second motor to be higher than the first motor, giving it a left movement.

4.1.3.2.3 Right movement

When the instruction left is received from MATLAB, the PIC program sets the ports RD0-RD3 the same as the forward movement. This time opposite to the left instruction, the CCP_1 is set at a higher value than CCP_2. This causes the speed of the first motor to be higher than the second motor, giving it a right movement.

4.1.3.2.4 Stop command

When receiving the stop command from MATLAB, the PIC program sets all ports RD0-RD3 to low and both CCP_1 and CCP_2 to low. This disables both motors causing the robot to stop.

4.1.4 *MATLAB Object Tracking*

The object tracking program is designed to first identify a reference object to be tracked beforehand. User is instructed to place the reference object in front of the camera with a plain background. The user then has to crop the image to fit the specified object and this reference image will be saved and used during the process of object tracking. In this project, the objects used are simple objects with least features or dimensions as the program uses shape as reference for tracking.

4.1.4.1 Edge detection

Edge detection marks the points in a digital image at which the luminous intensity changes sharply. Edge detection of an image filters out irrelevant information thus reduces significantly the amount of data, preserving the important structural properties of the target image. This technique is useful to detect the shape of given image.

Edges may be *viewpoint dependent* as edges may change as the viewpoint changes, which reflect the geometry of the object. In this project, in order to eliminate the viewpoint factor, a sphere shaped object is used for studying and experimental uses. In two dimensions, and higher, the concept of perspective projection has to be considered for different objects.

Function *edge* is used to apply this method.

A reference image (a ball) is used for edge detection and the similar object can be detected in the range of frames detected from the real time video captured using VCAPG2 in MATLAB.

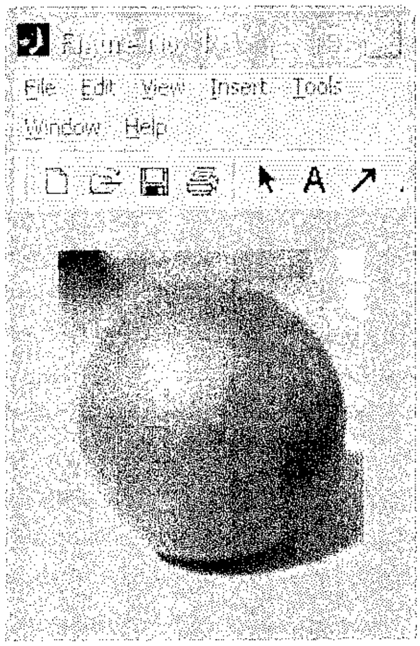


Figure 14 Reference Image for Object Tracking

BW2 = edge(I,'canny');

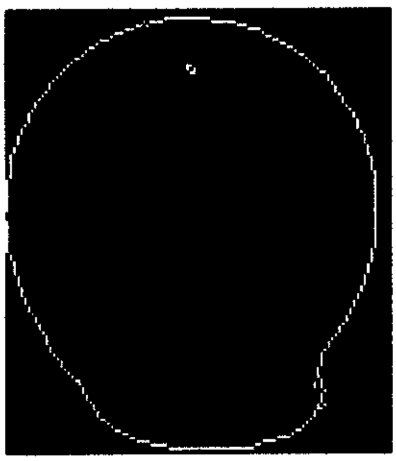


Figure 15 Edge Detection Result

Figure 16 shows a frame in which the ball is to be detected:

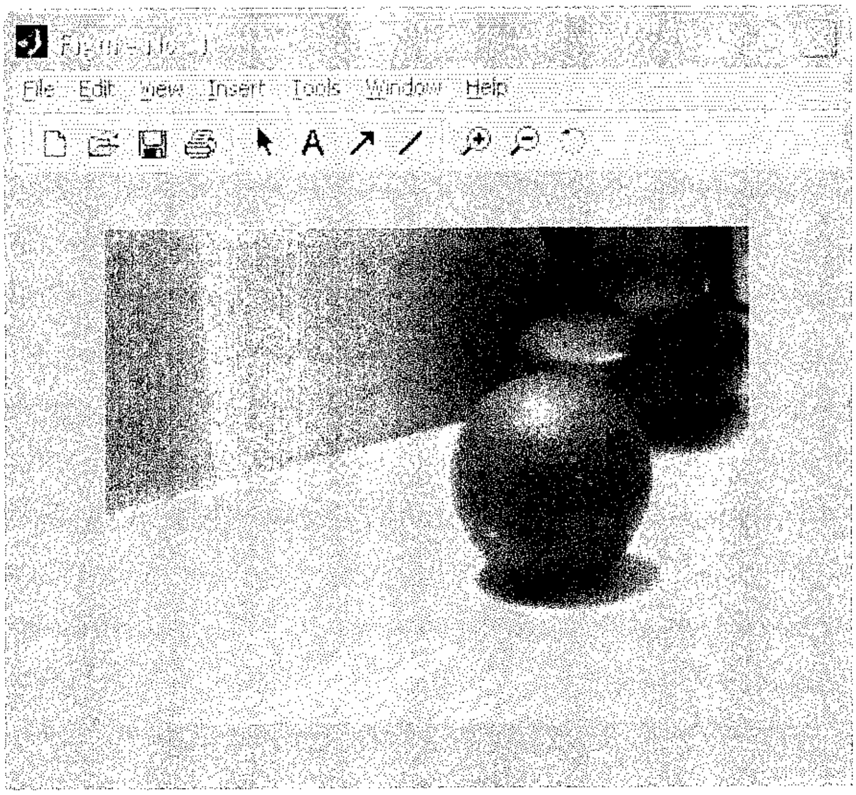


Figure 16 Input Image

The edge detection result is below:

Figure 17 Edge Detection Result of Input Image

From this, correlation can be done with less outside factors that can cause inaccuracy in the result.

4.1.4.2 Correlation

Filtering is done (using correlation) to find the similarities, and the maximum value of pixel (intensity from a binary image) is determined.

The position of the object is determined by the maximum value of pixel of the filtered result. This reference point is used to determine the area of the object tracked within the image, by using the function *roipoly*, from which the inside area of the indicated coordinates are turned white, and areas outside are turned black. The size of this area is determined by the size of the reference image itself.

The image created by the *roipoly* function is turned into 3-dimensional, and then multiplied with the original input image to be shown as the final result image.

A few experiments are done using edge detection and filtering of images. A reference image is as shown previously, which is a spherical object in this case (squash ball) and this reference image is correlated against images coming from the webcam.

4.1.5 RS 232 Connection with MATLAB

The function *fopen* can be used to connect to the serial connection RS232, in which case, this project would be using a USB port, connecting to USB-RS232 converter and direct to the PIC. Binary data would be used in this application.

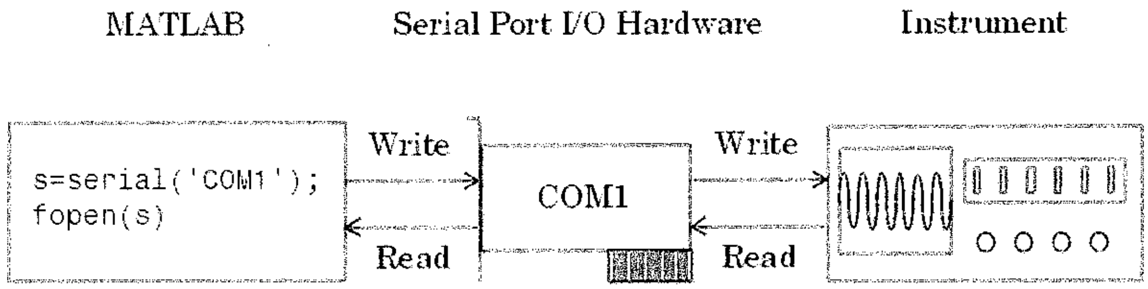


Figure 18 Serial Communication with MATLAB

4.1.6 Stereo Vision

In application of the stereo vision concept, a disparity map or "depth map" image is an efficient method for storing the depth of each pixel in an image. Performing this on the MATLAB software can determine the depth information of each pixel. A demonstration of disparity was done in [12], which analyses two photos taken in a room a foot apart to create a crude stereo image pair. Histogram equalization is performed to obtain the disparity map. Corresponding points (such as the corner of an object) on both images are found. Objects nearer will have greater separation (this is the disparity), and objects very far away will line up very close (they will have less disparity).

This experiment shows how stereo vision algorithms work. Every pixel in an image is matched with every pixel in the other using a correspondence algorithm. The depth of the pixel in the images can be viewed using a gray level histogram, where lighter objects are closer and darker objects are further. The disparity values at each pixel of both images are referenced to one image. The images below show the demonstration done in this experiment in [12].

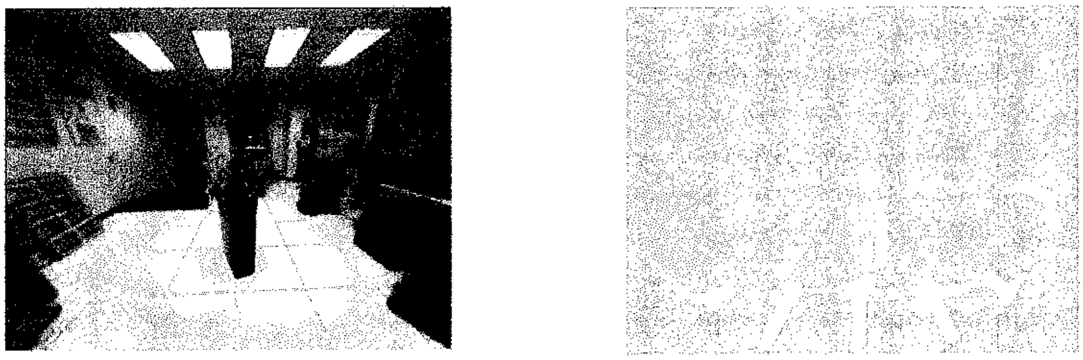


Figure 3. Indoor image of garbage can.

Figure 4. Histogram equalized disparity map.

Figure 19 Demonstration of disparity map

This method can be inaccurate due to occlusions and poor texture but is proven a reliable method. Through research done, quite a number of works done on computational stereo vision are done using disparity maps, including [15] and [16]. In terms of optimizing the process object tracking, the pixel depth information adds another dimension to direction of movement. The distance of the object tracked moves in a forward direction may be determined, which gives information to the mobile robot how far forward it needs to move.

4.2 System Functionality

4.2.1 Experiments Done with Robot Base

In the first stage of building the robot base, the first circuit built to check the movement algorithms is the H-bridge motor driver connected to the 3V motors, which drives the gearbox. This circuit is shown in Figure 10. In the first experiment of the movement algorithm, one of the wheels is disabled for the left or right movement. This experiment was implemented using push buttons to demonstrate the functionality of the robot. Table 5 shows the robot movement algorithm used in this experiment.

Table 5 :First experiment of robot movement algorithm

Input 1	0	1	0	0
Input 2	1	0	1	0
Motor1 Direction	Clockwise	Anti-clockwise	Clockwise	Disabled
Input 3	0	1	0	0
Input 4	1	0	0	1
Motor2 Direction	Clockwise	Anti-clockwise	Disabled	Clockwise
Robot Direction	Forward	Backwards	Left	Right

Once the first stage is functioning, the next stage is to build the controlling circuit from the microcontroller. This circuit is shown in figure 11. When experimenting with the PIC, a few programs are developed with port B as input, which is input from push buttons. Below is a sample of one of the programs used for the purpose of experiments. Port B is used as input from push button switches and Port D is used for input for H-bridge IC.

```
#include "16f877.h"

#USE DELAY (CLOCK=4000000) /*using a 4 Mhz clock
#FUSES XT, NOWDT, NOPROTECT, NOLVP
/*use  xt  mode,  no  watch  dog,  no  code  protect,  no  low  voltage
programming*/

char b;

void main()
{
set_tris_B(0xff) ;
set_tris_D(0x00) ;

while(1)
{

b = input(PIN_B0); /*forward movement*/
if (b==1)
       output_low(PIN_D4);
       output_low(PIN_D6);
       output_high(PIN_D5);
       output_high(PIN_D7);
       output_high(PIN_D2);
       output_high(PIN_D3);


   b = input(PIN_B1); /*backward movement*/
if (b==1)
       output_low(PIN_D5);
       output_low(PIN_D7);
       output_high(PIN_D4);
       output_high(PIN_D6);
       output_high(PIN_D2);
       output_high(PIN_D3);

b = input(PIN_B2);/*left movement*/
if (b==1)
       output_high(PIN_D6);
       output_low(PIN_D4);
       output_low(PIN_D5);
       output_low(PIN_D7);
       output_high(PIN_D2);
       output_high(PIN_D3);

b = input(PIN_B3);/*right movement*/
if (b==1)
       output_high(PIN_D4) ;
       output_low(PIN_D5);
       output_low(PIN_D6);
       output_low(PIN_D7);
       output_high(PIN_D2);
       output_high(PIN_D3);

b = input(PIN_B4);/*stop_brake*/
if (b==1)
       output_low(PIN_D3);
       output_low(PIN_D2);


}
}
```

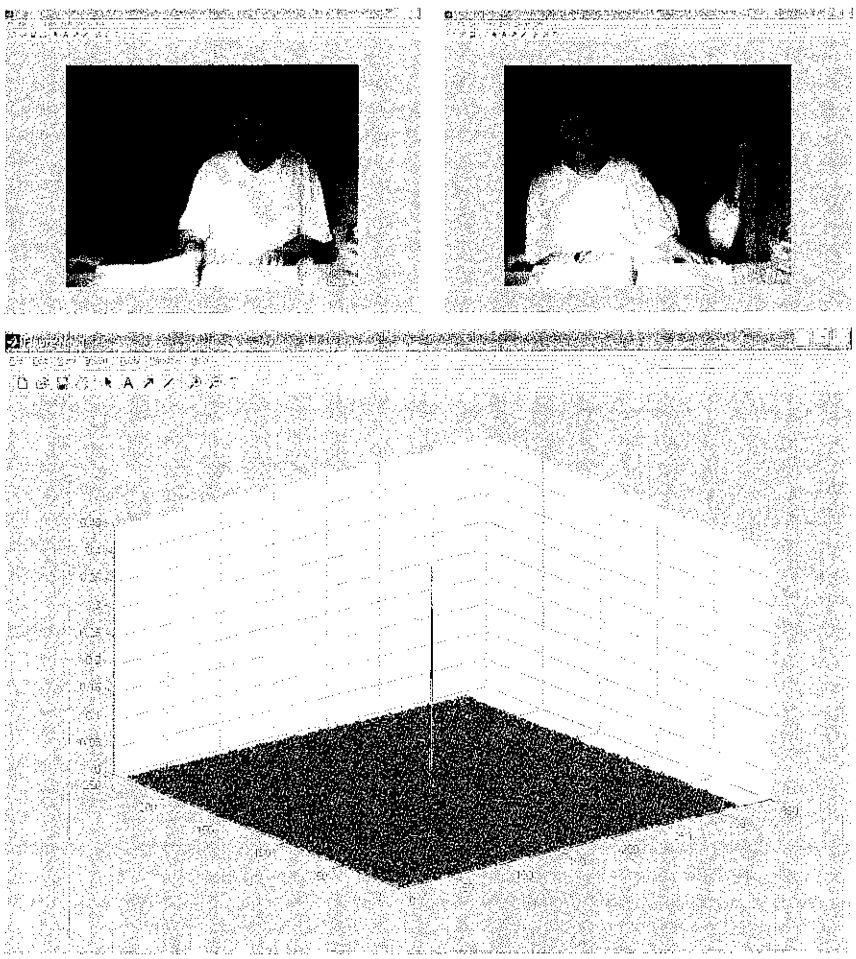
Here, CCP_1 and CCP_2 is not used for enable pins on the H-bridge IC. Enable A and B are set to high.

4.2.2 *Experiments Done with Object Tracking*

Initially, the object tracking applied in this project focuses on implementing image correlation using MATLAB. Two methods were explored; phase correlation and normalized cross correlation.

4.2.2.1 Phase Correlation

Phase correlation involves correlating two images, the reference image, and the shifted image in frequency domain. Two dimensional Fourier transform is done on each image. The peak of the phase correlation represents the location of the target object within the image. The results of phase correlation are the peaks locating the position of the shifted object in the targeted image. The next step would be to identify the peaks of the correlated images. Figure 20 shows the result for phase correlation.



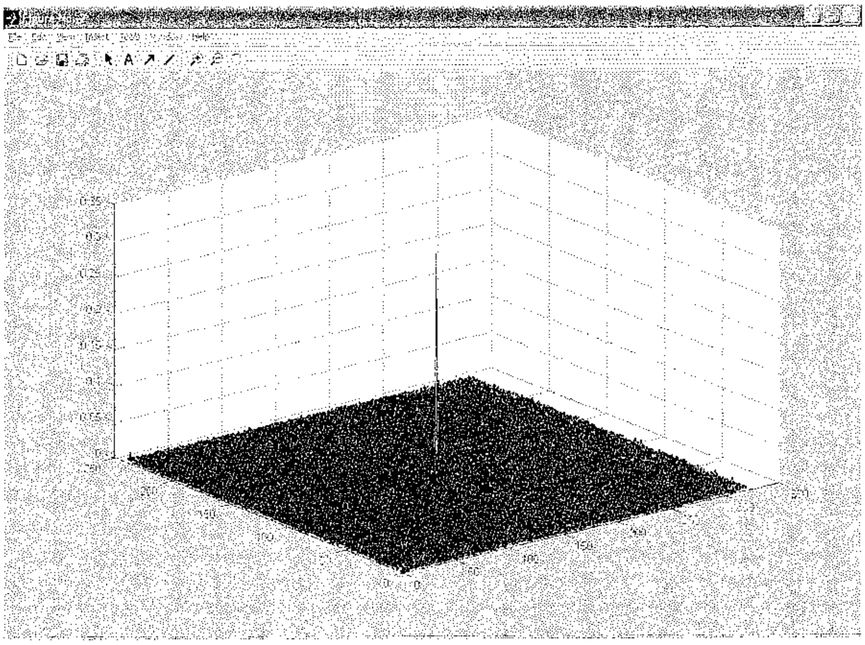


Figure 20 Phase correlation results

4.2.2.2 Normalized Cross Correlation

Normalized cross correlation is practiced and it is found to be reliable. This method requires a shift of the peak in the correlation map to be determined. The location is shifted by subtracting the coordinate with half of the reference image's dimension. The results show the peaks of the shifted image as shown in Figure 21.

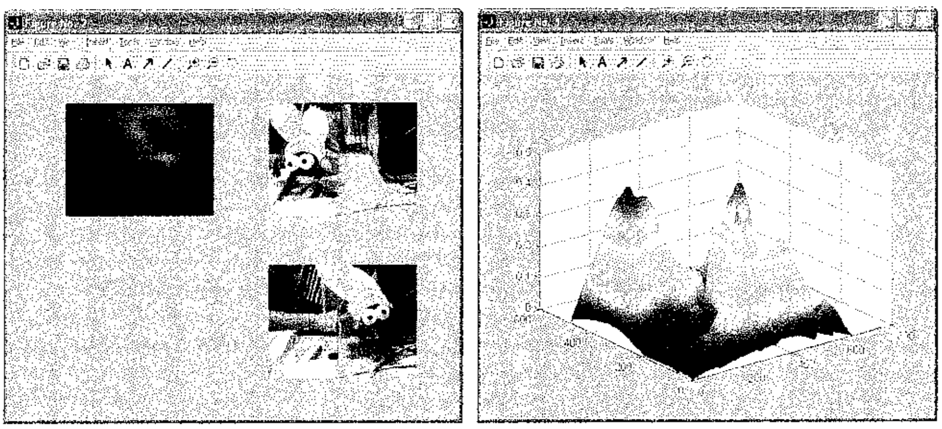


Figure 21 Normalized cross-correlation results

From the experiments done with both phase and normalized cross correlation, it proves that movement of certain objects can be detected from one point in the screen to another. The problem with using this method is that, it uses up a lot of computational time which interferes greatly with the real-time object tracking of this project. Both methods lack in fast responses when tracking. A problem can also be detected when identifying a reference object to be tracked. The phase correlation and normalized cross correlation methods are reliable when identifying the motion and the position of a moving object accurately, however, given the condition and objective of the project, more emphasis need to be put on fast response, and not accuracy. There is also a factor of identifying motion from a mobile image acquisition tool, which is the webcam. Therefore a new method is explored, which is edge detection and filtering, which focuses more on identifying a referenced object within a screen instead of detecting motion.

4.2.2.3 Edge Detection and Filtering

In this project, a few experiments were done using a few simple objects to test the functionality of the algorithm. The first experiment done is with a sphere object which is a squash ball. Since using a sphere object, with fewer dimensions to consider and with a distinctively dark colour eliminates the elements of noise from the surrounding environment, it helps to simplify the process of developing the object tracking algorithm. Later on, objects with more dimensions are experimented on, such as a cube or even a flower shaped object.

Tests were done to verify the timing of the object tracking process using the function *tic* and *toc* which acts as a stopwatch. It times the amount of time the process of the program to complete one cycle of object tracking. Table 6 shows the results of the tests done.

Table 6 :Time test result for object tracking using edge detection

Position of object (in frame)	Test 1	Test 2	Test 3	Average time
Left	0.8210s	0.8110s	0.8020s	0.8113
Right	0.8510s	0.8110s	0.8310s	0.8310
Middle	0.7920s	0.8310s	0.8320s	0.8183

From these tests, it can be seen that one cycle of object tracking process takes on average 0.8202 seconds, which provides the fast response needed for real-time taking. It can also be seen that the position of the object within the frame does not affect the time taken to complete one cycle of object tracking.

4.2.2.4 Experiment with different shaped objects

4.2.2.4.1 Sphere Shaped Object

The reference image (squash ball) is shown in Figure 22.

Figure 22 Reference Image 1

Some results showing the referenced object detected within the frames of the webcam are shown in Figure 23.

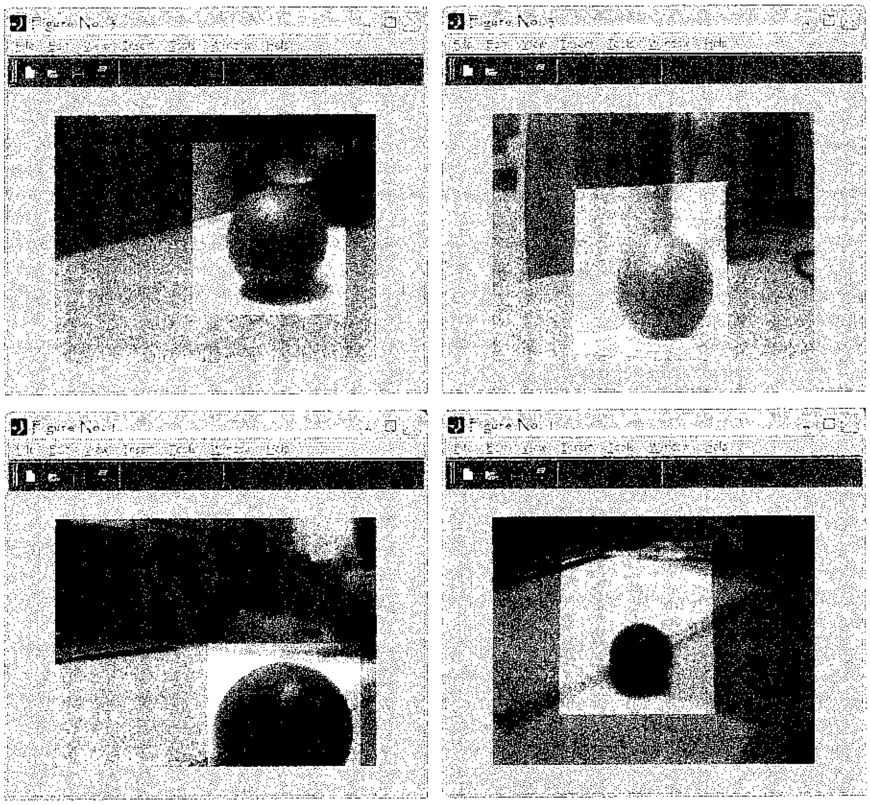


Figure 23 Result of object tracking with a sphere object

However, under certain conditions when the distance of the object from the camera is large or in the case of motion of camera or the object itself, the program is unable to track the referenced object. This is shown in Figure 24.

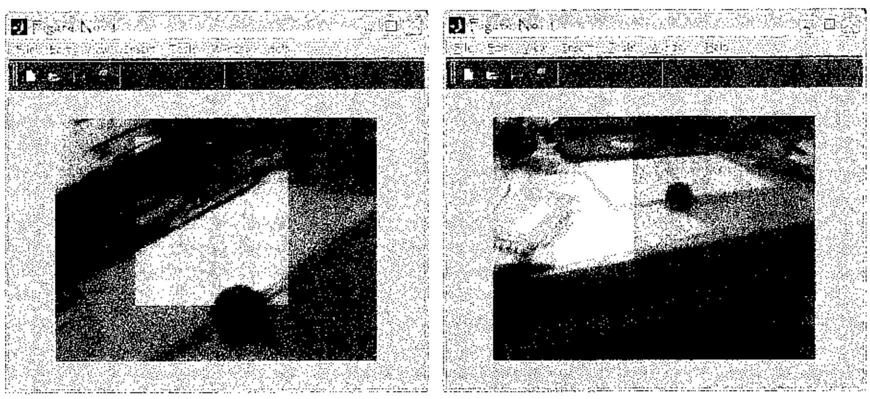


Figure 24 Result of object not tracked under different conditions

4.2.2.4.2 Cube Shaped Object

Figure 25 shows the second reference image used in the experiment which is a cube box.

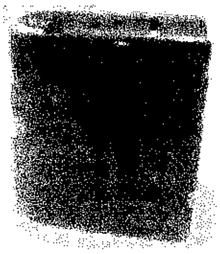


Figure 25 Reference Image 2

This object is similarly distinctive to the previous object in terms of colour. It is dark in colour which simplifies the process of tracking in a light environment. Figure 26 shows some of the results obtained from the experiment.

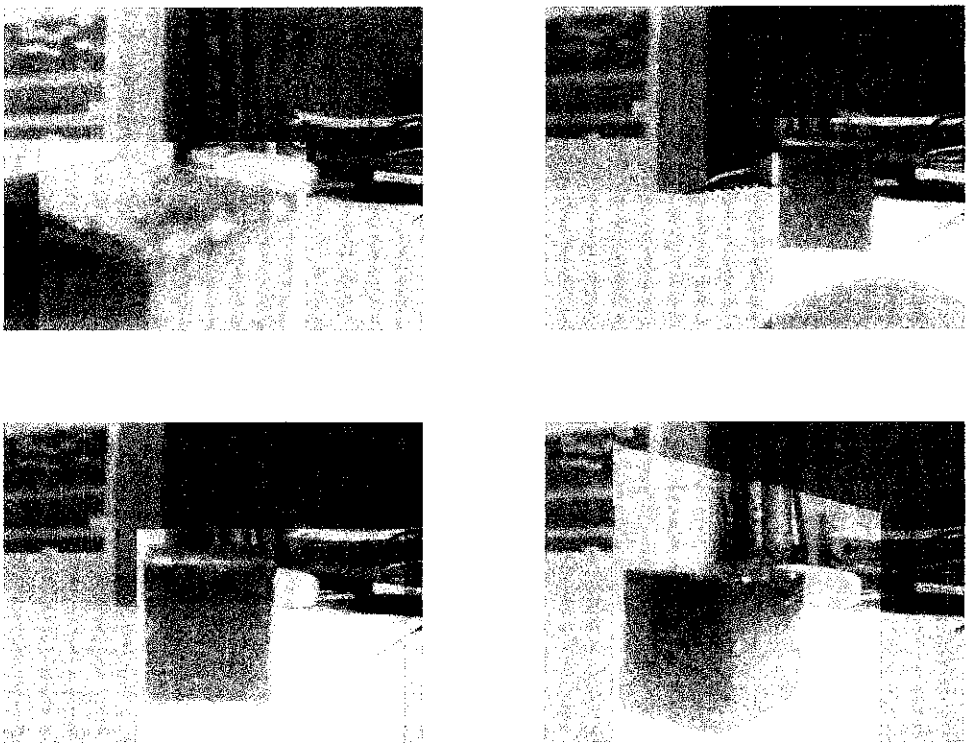


Figure 26 Result of object tracking with a cube object

It can be seen from this experiment, the object can be tracked around the frame. This is due to the distinctive colour and shape of the reference object from the surrounding environment.

4.2.2.4.3 Flower Shaped Object

In this experiment, an object which is flower shaped is used as a reference image. The objective is to determine whether the characteristic of the object, which is different from surrounding background can be successfully detected, taking into consideration of the viewpoint angles.

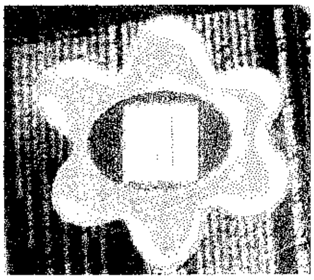


Figure 27 Reference image 3

The results are shown in Figure 28.

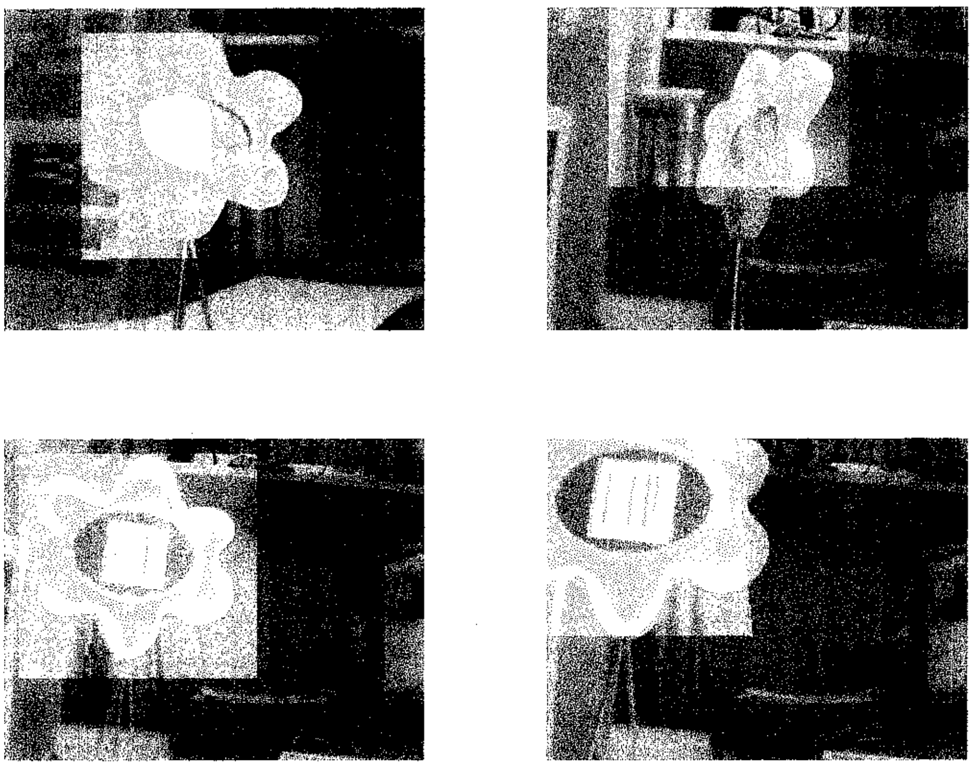


Figure 28 Result of object tracking with flower shaped object

As seen from the results in Figure 28, the object is successfully tracked in the frame given a controlled condition. This is due to the difference of shape and colour of the object from the surrounding background.

From the experiments above, it can be seen the algorithm created using edge detection and correlation has detected the reference object in given input images. However, the accuracy differs under certain conditions; when object is at a distance from camera or under moving conditions (either moving object or moving camera). It can be seen that objects are tracked when given a condition in which the background and the referenced object are in contrast with each other, in which case, this object tracking algorithm works best in an open space.

4.3 Integration between Hardware and Software

The results from object tracking in MATLAB are used for the input control of the PIC. The position of the object in the frame or image determines the direction of the movement of the robot. This is achieved by dividing the frame into three sections, namely left, middle and right. The frame size of the webcam is 320 by 240 pixels. When dividing it into three sections vertically, it gives a range of pixels **0-107** for left position, **107-214** for middle position and **214-320** for right position. The result of *filter2* returns the position of maximum pixel value from where the input image is most similar to the reference object. This pixel position can now be categorized as being in the left, right or middle section of the frame (Figure 28); therefore can send the information needed to the PIC via RS232 serial communication.

Some experiments were done to determine the pixel position and verified with the actual position the object is detected/tracked in the frame. Table 7 shows some of the results.

Table 7 : Results for object tracking pixel positioning

	X-axis			Y-axis		
	X1	X2	X3	Y1	Y2	Y3
Left	92	100	83	104	103	102
Right	235	240	239	93	90	88
Middle	155	154	145	101	96	94

In this experiment, the Y-axis is not taken into consideration. Only X-axis is used for the left, right or straight directions. As shown in table 7, three measurements are taken for each region. It can be seen that for the left region, all the pixel x-coordinates are **below 107**. For the right region, all pixel positions are **larger than 214** and for the middle region, all pixel positions are **between 107 and 214**. This verifies that this method of identifying the position of object is true, in terms of sending instructions to the PIC. Below are some MATLAB codes used to send information to the RS232 using the function *fopen*.

```
s = serial ('COM1');
fopen(s)
.
.
.
.

%if object is in the left region then send char 'l'
if ( pcolumn < 107 )
    fprintf(s,'l');

%if object is in the right region then send char 'r'
else if ( pcolumn > 214 )
    fprintf(s,'r');

%if object is in the middle region then send char 'f'
else if(107 <= pcolumn <= 214)
     fprintf(s,'f');

%if object is not detected or returns empty arrays, send char 's' for
stop
else
     fprintf(s,'s');
end
```

4.4 Problems Faced

1 Cameras need to be mounted on top of castor unit for stability

2 Smaller and wireless cameras need to be utilized

3 Wireless connection for data transmission need to be implemented for better mobility

4 Stereo vision need to be implement where frames from both cameras are correlated simultaneously

5 Higher gearbox ratio need to be implemented with lower rotation speed as reverse movement is hard to be done without an initial push

6 Object tracking need to be done using different shapes with different viewpoints and also different lighting.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This project has the objective of developing a mobile robot with stereo vision object tracking capabilities and has the scope of studies under computer vision, microcontroller and robotics. The mobile robot has the capabilities of moving in four directions manually controlled. The final robot base is built accordingly and tested with the circuitries for troubleshooting. The object tracking algorithm is developed that it is able to detect shapes of objects in comparison with a reference object using the edge detection technique. A new method is also explored with object recognition in videos utilizing text retrieval method which helps with the factor of difference of viewpoints (angles) which changes the shape of object. The stereo vision algorithm is researched and this helps in the accuracy of determining the position of object tracked.

A full algorithm for the robot movement is completed, undergoing through a series of experiments using the PIC. The experiments are done step by step, integrating the series of circuits as the project develops, from basic motor controlling function, merely using push button switches, to sending data from the computer to the PIC through serial communication, RS232. This project focuses on simplifying the object tracking methods and the robot mobility mechanism to save processing and computational time to achieve real-time tracking. This is achieved by using DC motors, with a gearing system for robot mobility and edge detection for object tracking. Experiments were done to ensure time efficiency of the object tracking and swift movements of the mobile robot.

5.2 Recommendations

Stereo vision is a field that is heavily researched nowadays. The concept has been researched in this project but has not been implemented as part of the system, therefore, as a recommendation of further development of this project would be to further develop the concept of stereo vision on the mobile robot. Object tracking done in this project needs to be optimized for better accuracy as emphasis has been placed on fast response of the system.

The current mobile robot works based on a wired serial connection, from camera to PC and PC back to the PIC on the robot. Further development need to be placed upon the mobility of the mobile robot, possibly using wireless connection for data transmission, from PC to mobile robot.

REFERENCES

Books

[1] Richard Bartnett, Larry O'Cull and Sarah Cox, *Embedded C Programming and the Microchip PIC,* Thomson Delmar Learning, 2004

[2] M. J. Tarr and H. H. Bulthoff, *Object Recognition in Man, Monkey, and Machine,* The MIT Press, 1999

[3] M. Shah and R. Jain, *Motion-Based Recognition, Computational Imaging and Vision,* Kluwer Academic Publishers, 1997

[4] R. C. Gonzalez, R. E. Woods and S. L. Eddins, *Digital Image Processing using MATLAB,* Pearson Prentice Hall, 2004

[5] Ulrich Nehmzow, *Mobile Robotics: A Practical Introduction,* 2nd Edition, Springer, 2003

[6] C. Ray Asfahl, *Robot and Manufacturing Automation,* 2nd Edition, John Wiley & Sons, Inc, 1992

[7] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision,* 2nd Edition, Cambridge University Press, 2003

Reports/ Thesis

[8] Fadliyanto, "Pan and Tilt Platform for Object Tracking Utilizing Webcam", December 2006

[9] Munira N. F. Ishak, "Simple Object Motion Detection and Speed Estimation", Digital Signal Processing, May 2006

[10] Sivic, J. and Zisserman, A., "Video Google: A Text Retrieval Approach to Object Matching in Videos", Proceedings of the International Conference on Computer Vision (2003)

[11] Junji Yamato, "Tracking Moving Object by Stereo Vision Head with Vergence for Humanoid Robot", Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1998.

[12] Donald Rosselot and Ernest L. Hall, "Processing real-time stereo video for an autonomous robot using disparity maps and sensor fusion", Center for Robotics Research, Department of Mechanical, Industrial, and Nuclear Engineering, University of Cincinnati

[13] Paul Withagen, "Stereo vision: Lab Exercise 4 for the Computer Vision Course", University of Amsterdam, The Netherlands, 2002

[14] S.Bahadori, L.Iocchi, D.Nardi, G.P.Settembre,"Stereo Vision Based Human Body Detection From a Localized Mobile Robot", Universit` di Roma La Sapienza, Roma, Italy, 2005.

[15] M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in computational stereo", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 25, No. 8, August 2003

[16] D. Scharstein, R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", International Journal of Computer Vision, Vol. 47(1/2/3), pp 7–42, 2002.

Others

[17] Definition of Robotics, New Oxford Dictionary of English

[18] *http://www.mathworks.com/matlabcentral/fileexchange/loadCategory.do*

APPENDICES

APPENDIX A: PROGRESS/GANTT CHART

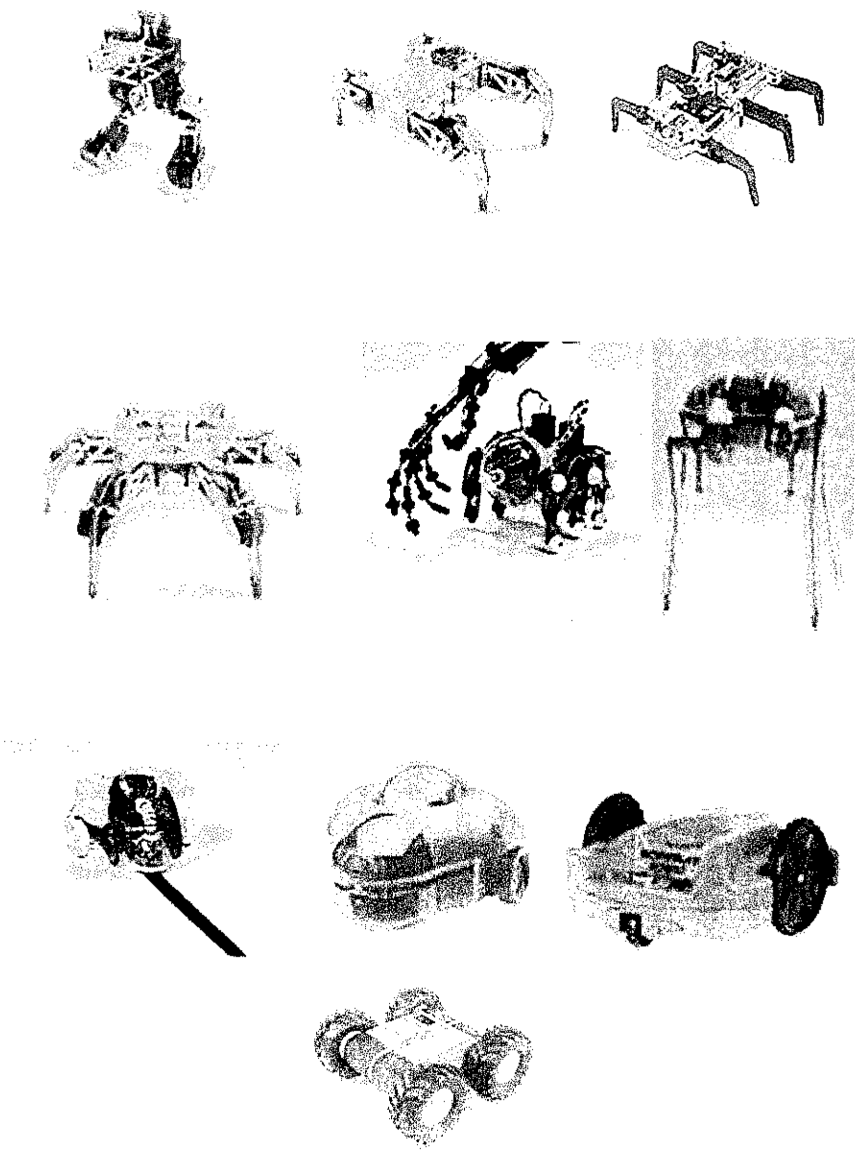
Final Year Project Gantt Chart Semester 1 January 2007

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
	-Propose Topic														
	-Topic assigned to students														
2	Preliminary Research Work														
	-Introduction														
	-Objective														
	-List of references/literature														
	-Project planning														
3	Submission of Preliminary Report				0										
4	Robot Design														
	-Reference/Literature														
5	Submission of Progress Report								0						
6	Image Processing Development														
	-Reference/Literature														
	-Experiments														
7	Circuit Implementation														
8	Hardware Fabrication														
9	Submission of Interim Report Final Draft											0			
10	Submission of Interim Report														0
11	Oral Presentation														0

Final Year Project Gantt Chart Semester 2 July 2007

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	Object tracking/recognition/correlation																					
	-research																					
	-algorithm																					
2	PIC Programming																					
	-work in laboratory																					
3	Submission of Progress Report 1				0																	
4	Mobile robot base																					
	-design																					
	-building																					
5	Webcam implementation onto robot base																					
	-testing and troubleshooting																					
6	Stereo vision implementation for object tracking																					
7	Submission of Progress Report 2								0													
8	Integration/connection between robot base with MATLAB																					
9	Troubleshooting of integration																					
10	Research on wireless connection																					
11	Final report and technical report writing																					
12	Submission of Dissertation Final Draft													0								
13	Submission of Project Dissertation Soft Cover															0						
14	Submission of Dissertation Technical Report																0					
15	Troubleshooting of prototype																					
16	Preparation for presentations and exhibitions																					
17	Oral Presentation																			0		
18	Submission of Project Dissertation Hard Cover																					0

APPENDIX B: VARYING ROBOT DESIGNS, LEGGED (ABOVE) AND WHEELED (BOTTOM)



APPENDIX C: PIC C CODES FOR ROBOT CONTROL

```
#include <16f877.h>
#USE DELAY (CLOCK=4000000) /*using a 4 MHz clock*/
#USE RS232 (baud=4800, parity=N, xmit=PIN_C6, rcv=PIN_C7, stream=,
bits=8 )
#use standard_io(B)
#use standard_io(D)
#FUSES XT, NOWDT, NOPROTECT, NOLVP
/*use xt mode, no watch dog, no code protect, no low voltage
programming*/


char b;
int en1, en2;

main()
{

set_tris_D(0x00) ;
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);

output_low(PIN_D0);
output_low(PIN_D2);
output_low(PIN_D1);
output_low(PIN_D3);


b = getchar();

while(1)
{

  en1 = CCP_1;
  en2 = CCP_2;

if (b=='f')                /*forward movement*/

       output_low(PIN_D0);
       output_low(PIN_D2);
       output_high(PIN_D1);
       output_high(PIN_D3);

       en1=100;
       en2=100;

if (b=='l')                /*left movement*/

       output_low(PIN_D0);
       output_low(PIN_D2);
       output_high(PIN_D1);
       output_high(PIN_D3);

       en1=100;
       en2=200;
```

```
if (b=='r')                    /*right movement*/

        output_low(PIN_D0);
        output_low(PIN_D2);
        output_high(PIN_D1);
        output_high(PIN_D3);

        en1=200;
        en2=100;



if (b=='s')                    /*motor_brake*/

        output_low(PIN_D0);
        output_low(PIN_D2);
        output_low(PIN_D1);
        output_low(PIN_D3);

          en1 = 0;
          en2 = 0;

}
}
```

APPENDIX D: MATLAB CODES FOR OBJECT TRACKING

```
%reference image acquisition
%--------------------------------------------------------------------------
pause (0.5);
clear all
clc

%initialising camera
vcapg2;
pause(2.5);
fprintf('\nCamera ready');

%capturing reference image
fprintf('\nPlease show reference object/image with a
plain background for 5 seconds');
ready = input('\n Ready? Type (y) then enter to
continue : ','s');
if (ready == 'y');
pause(1);
fprintf('\n5sec');
pause(1);
fprintf('\n4sec');
pause(1);
fprintf('\n3sec');
b=vcapg2(0);
pause(1);
fprintf('\n2sec');
pause(1);
fprintf('\n1sec');
pause(1);
else
    fprintf('\nPlease start again');
end

%cropping image
fprintf('\nPlease crop image using cursor, then close
window');
b=imcrop(b);
%saving reference image
fprintf('\nSaving reference image');
naming = input('\n Type the name of the new image file
(filename.ext) : ','s');
imwrite(b,naming,'jpg');
```

```
%checking reference image
I=imread(naming);
I=im2double(I);
sizI=size(I);
fprintf('\nShowing reference image');
imview(I);
I=rgb2gray(I);
J=graythresh(I);
BWI=im2bw(I,J);
BWI=bwperim(BWI);
pause (3.5);
imview(BWI);

correct = input('\nDoes image show the correct shape of reference image? (y/n): ', 's');
if (correct == 'y')
    fprintf('\nStart object tracking in 5 seconds');
    pause(1);
    fprintf('\n5sec');
    pause(1);
    fprintf('\n4sec');
    pause(1);
    fprintf('\n3sec');
    pause(1);
    fprintf('\n2sec');
    pause(1);
    fprintf('\n1sec');
    pause(1);
    fprintf('\nObject Tracking');
else
    fprintf('\nPlease restart program');
    break;
end
```

```
%object tracking
%------------------------------------------------------------------------


%opening serial port
s = serial ('COM1');
fopen(s)
while(1)
    T=vcapg2(0);
    T_G=rgb2gray(T);
    T=im2double(T);
    sizT=size(T);
    BW=edge(I);
    BW2=edge(T_G);
    out=filter2(BW,BW2);
    o=max(max(out));
    output=(1/o)*out;
    pixel=find(output==1);
    pcolumn=fix(pixel/sizT(1));
    prow=mod(pixel,sizT(1));
    rdis=fix(sizI(1)/2);
    cdis=fix(sizI(2)/2);
    cmin=pcolumn-cdis;
    cmax=pcolumn+cdis;
    rmin=prow-rdis;
    rmax=prow+rdis;
    c=[cmin cmin cmax cmax];
    r=[rmin rmax rmax rmin];
    m=roipoly(T,c,r);
    m=im2double(m);
    m=0.5*(m+1);
    mask(:,:,1)=m;
    mask(:,:,2)=m;
    mask(:,:,3)=m;
    final = mask.*T;
    imshow(final);

%determining    position    of    object    and    sending
instruction to PIC
   if ( pcolumn < 107 )
    fprintf(s,'l');
else if ( pcolumn > 214 );
        fprintf(s,'r');
    else if(107 <= pcolumn <= 214)
            fprintf(s,'f');
        else
            fprintf(s,'b');
        end
    end
end
end

fclose(s)
delete s
clear all
```

APPENDIX E: DATASHEET PIC16F877

MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

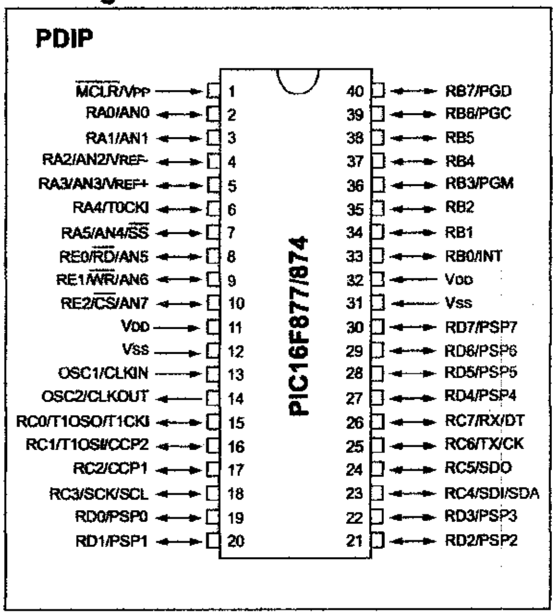
Devices Included in this Data Sheet:

- 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

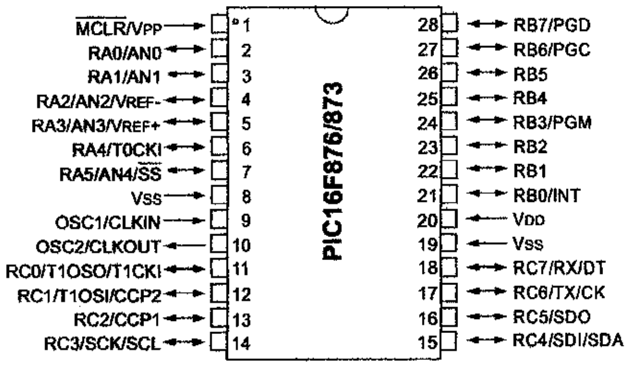


Peripheral Features:

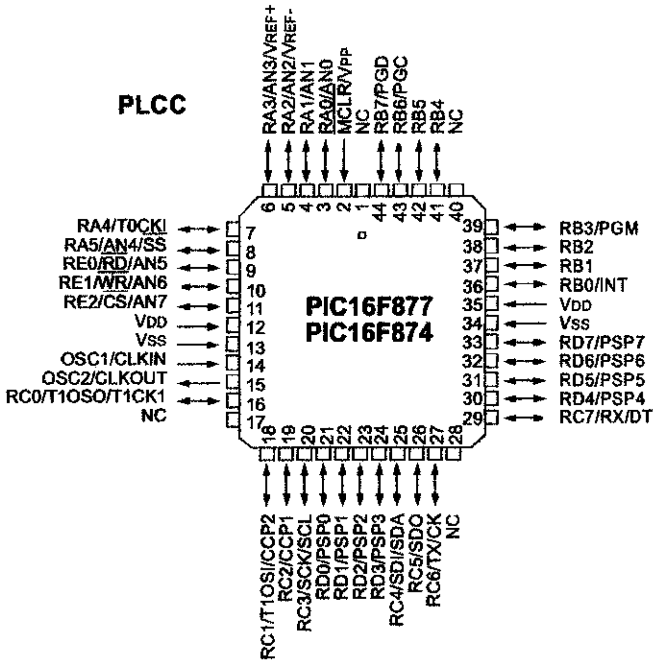
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. 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^{2}C™ (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

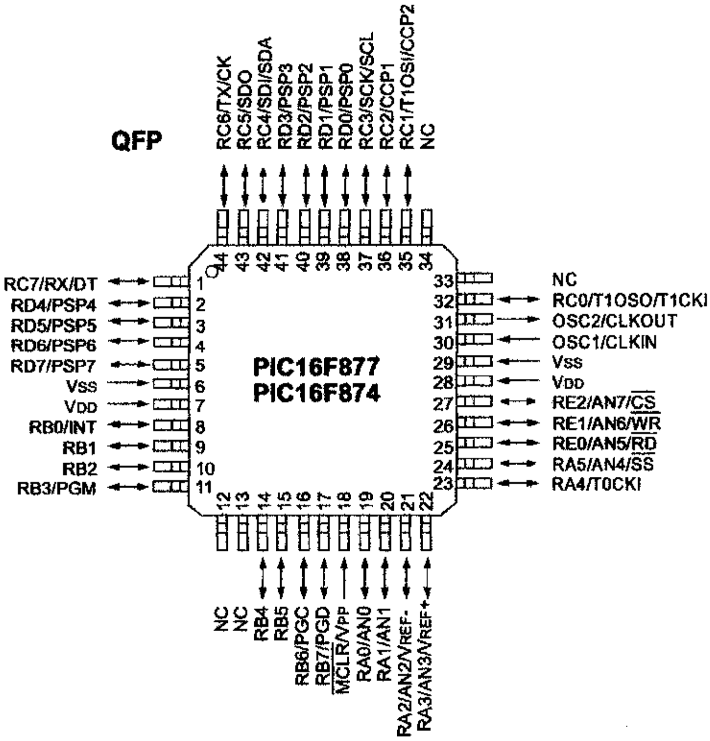
PDIP, SOIC



PLCC



QFP



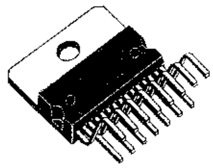
Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit 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

APPENDIX F: DATASHEET L298 DUAL FULL-BRIDGE DRIVER

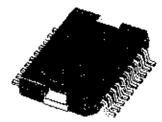
L298

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



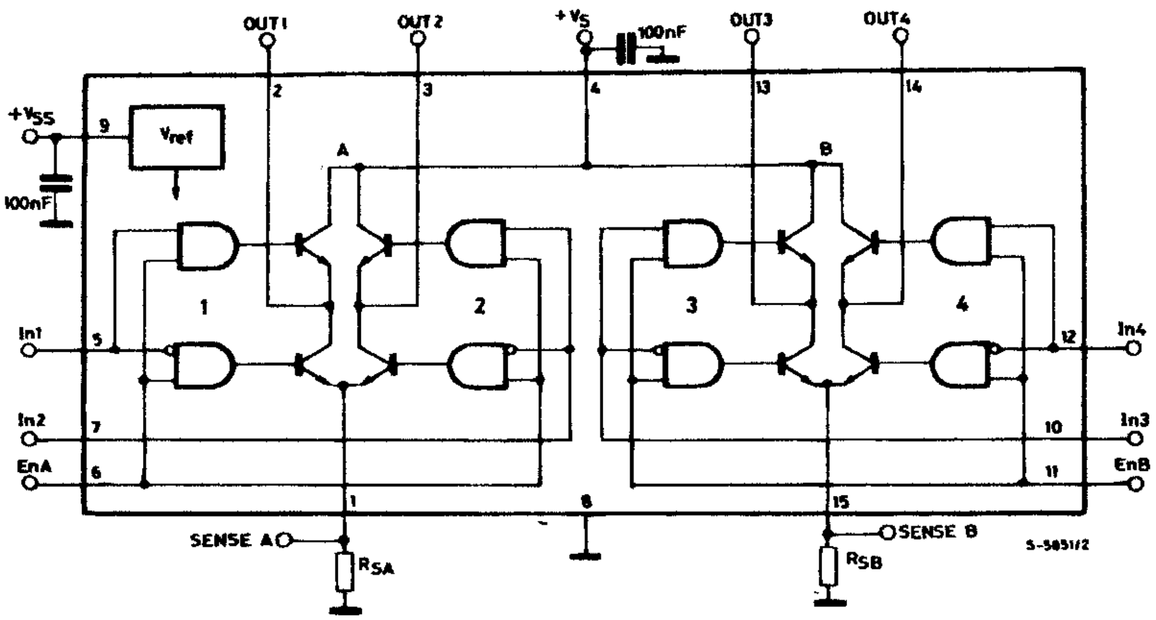
PowerSO20

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.

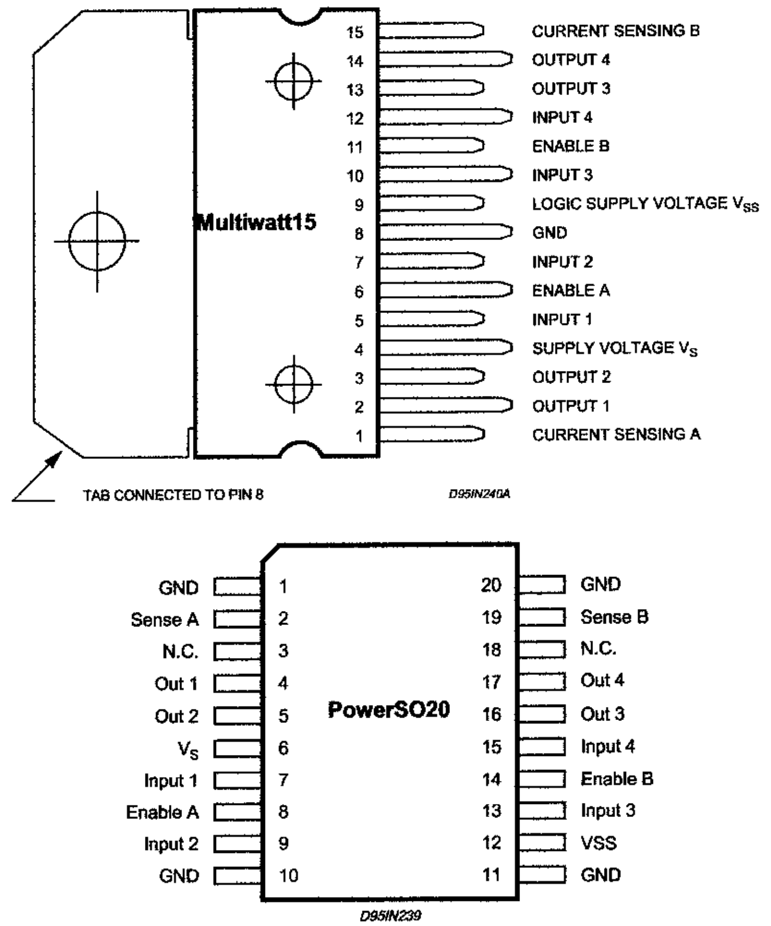
BLOCK DIAGRAM



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 $t \leq 50$ 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.

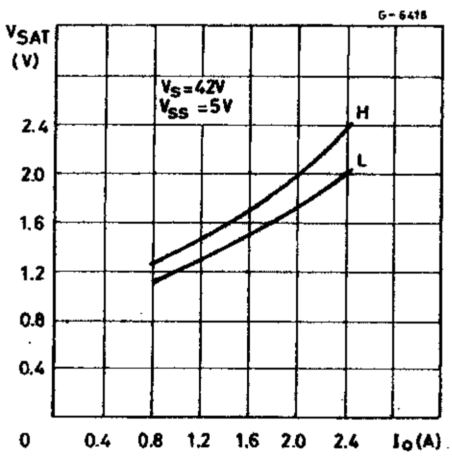
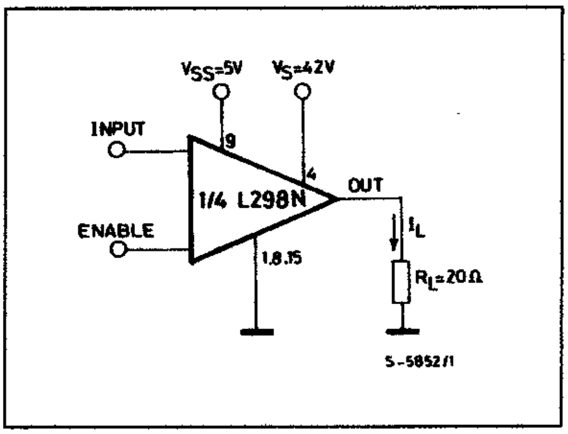


Figure 2 : Switching Times Test Circuits.



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

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

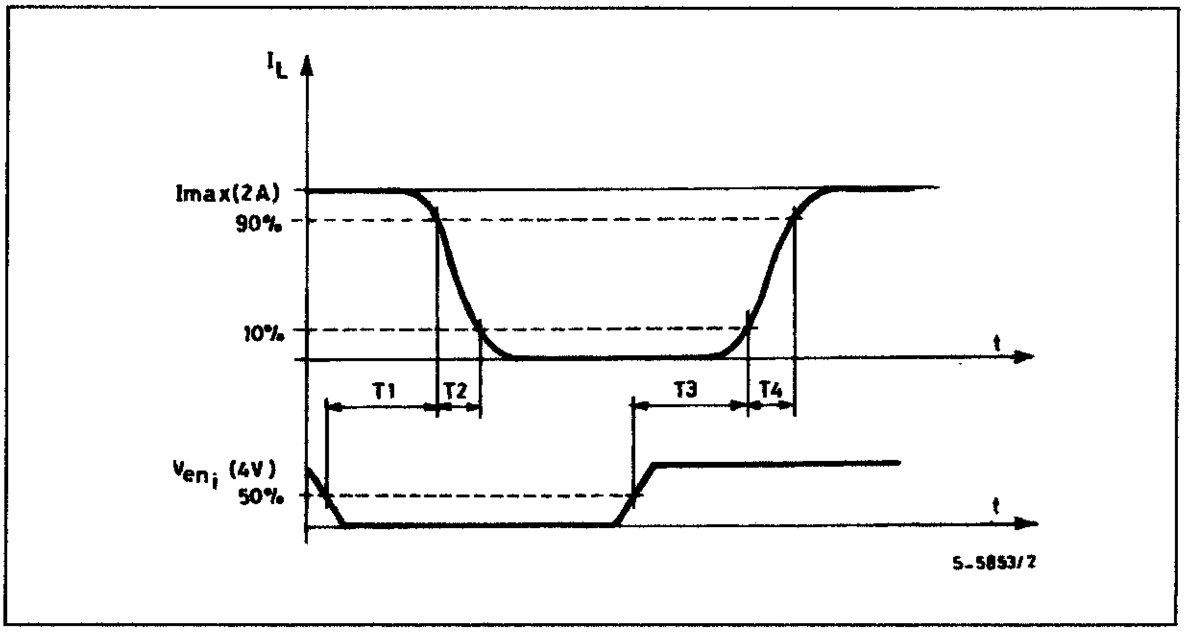
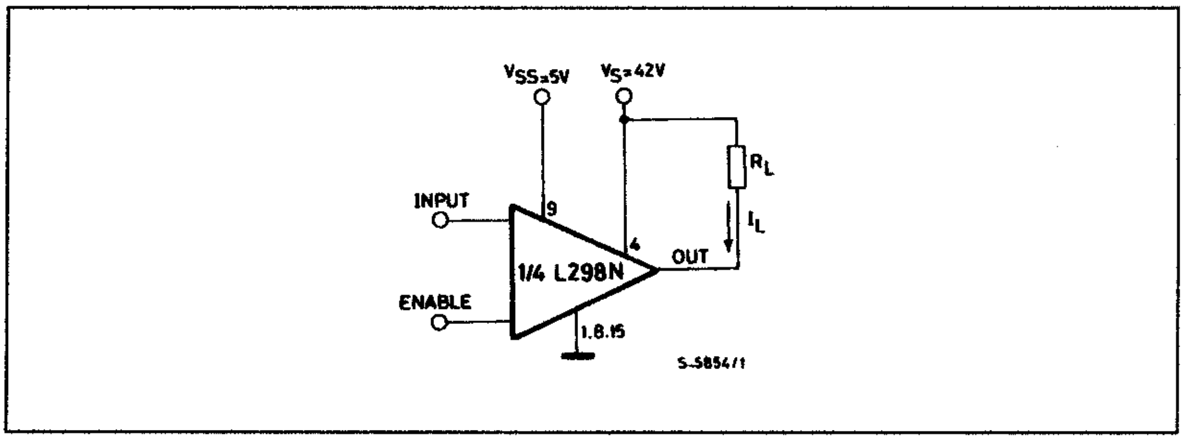


Figure 4 : Switching Times Test Circuits.



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.

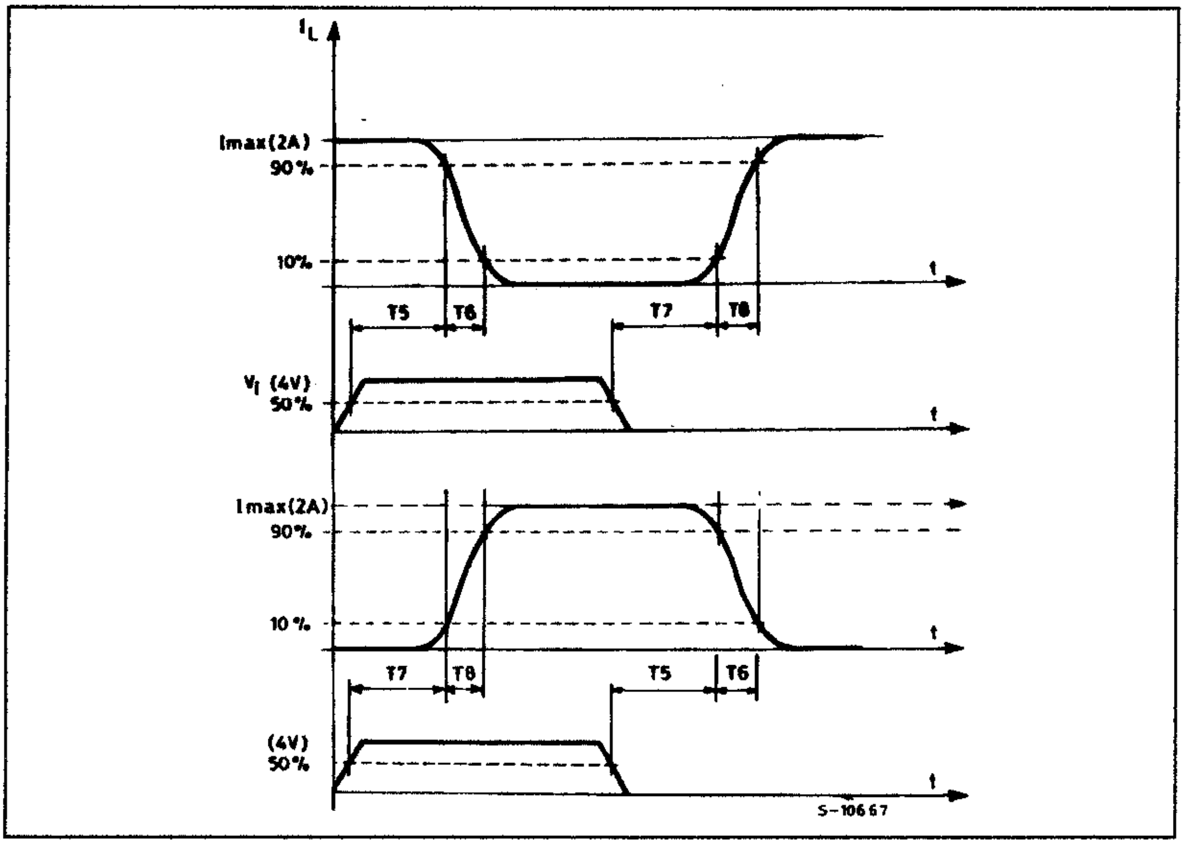
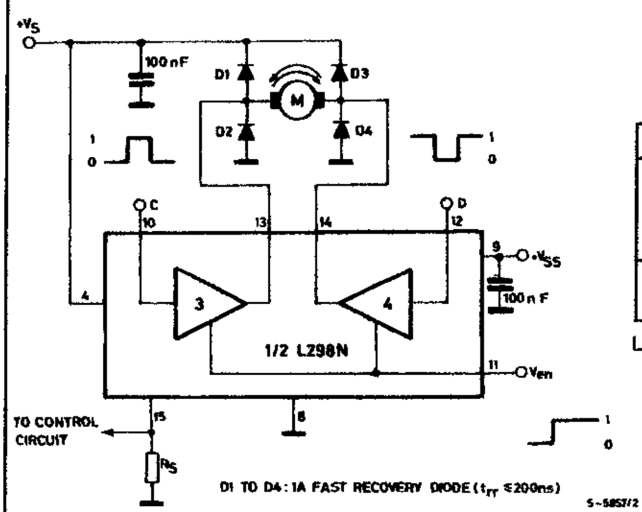


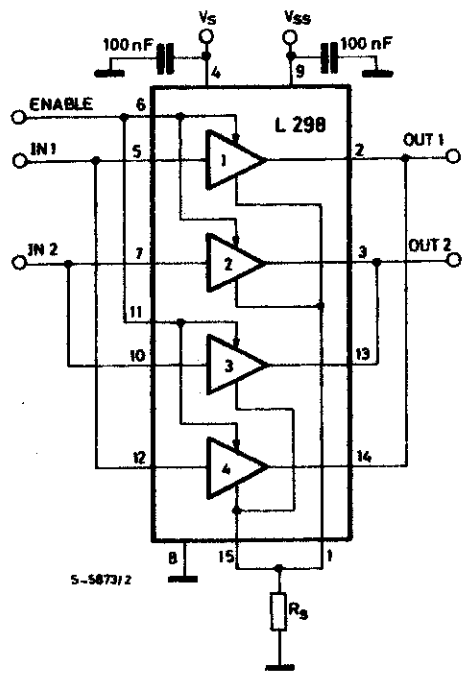
Figure 6 : Bidirectional DC Motor Control.



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.

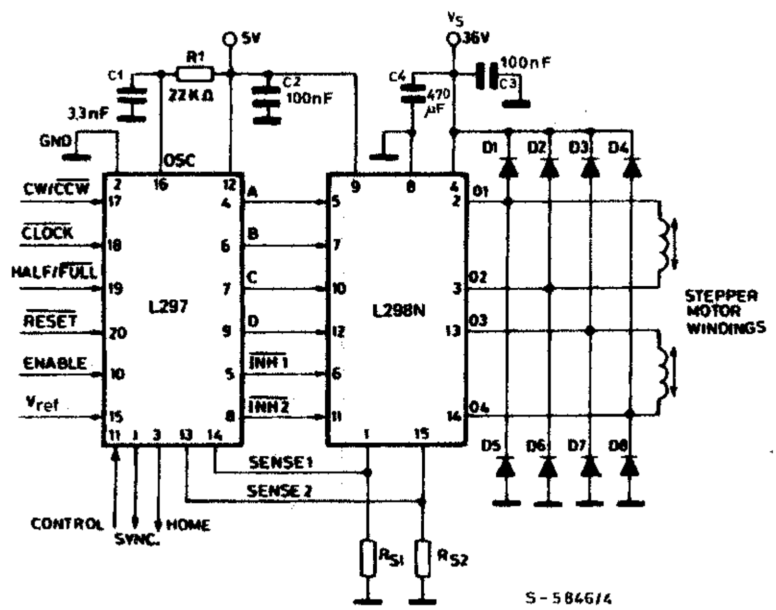
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

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 $\begin{cases} V_F \leq 1.2\ V\ @\ I = 2\ A \\ trr \leq 200\ ns \end{cases}$

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

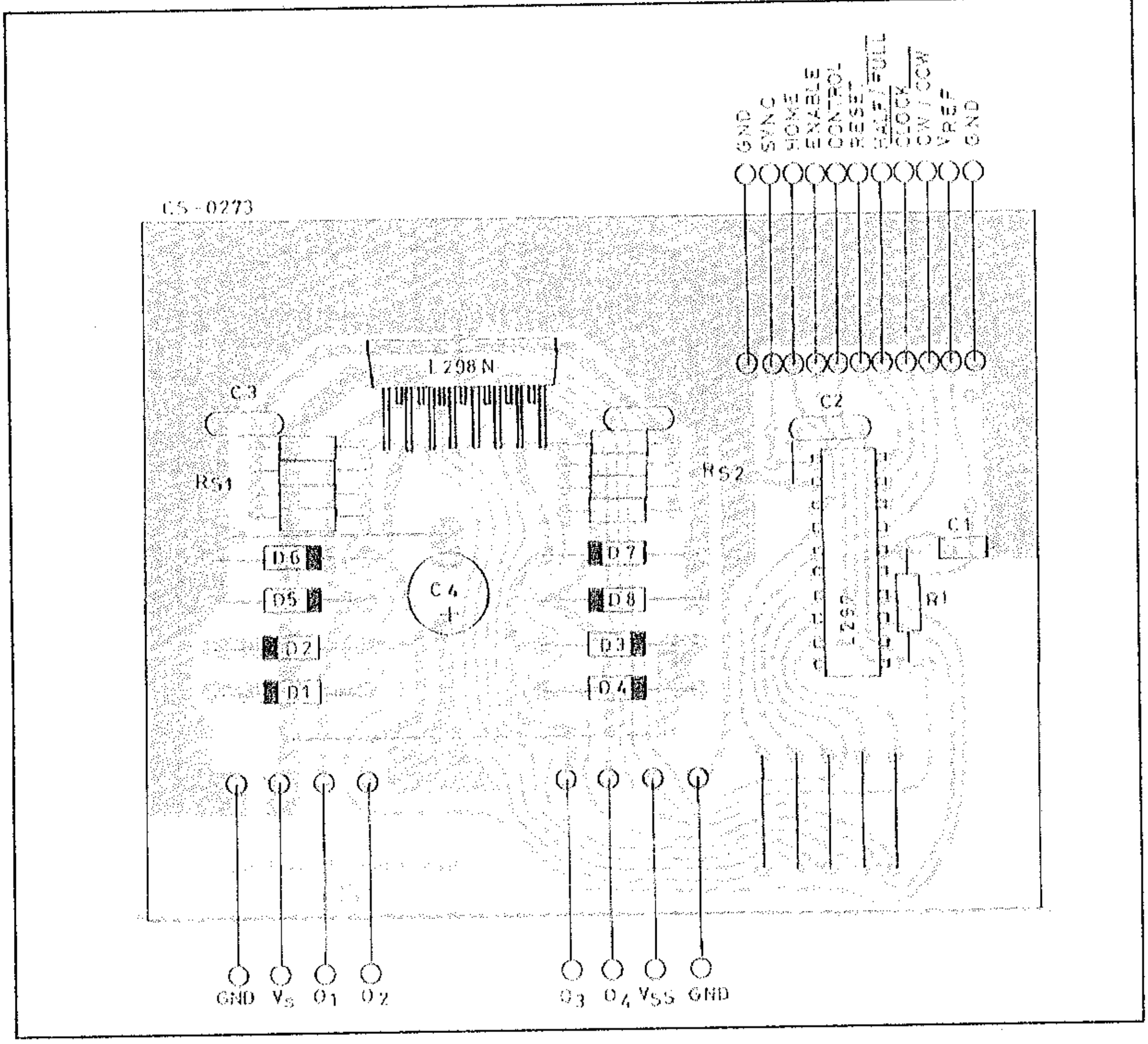
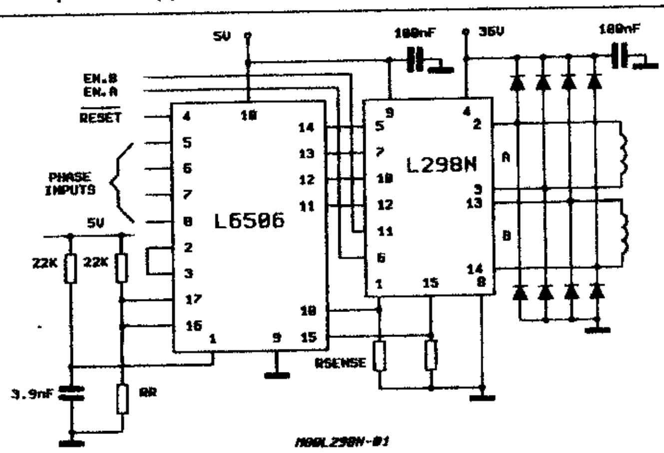


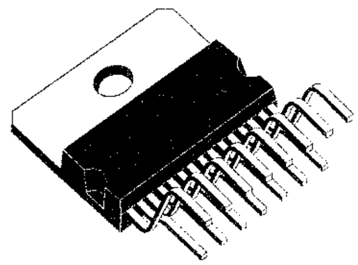
Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



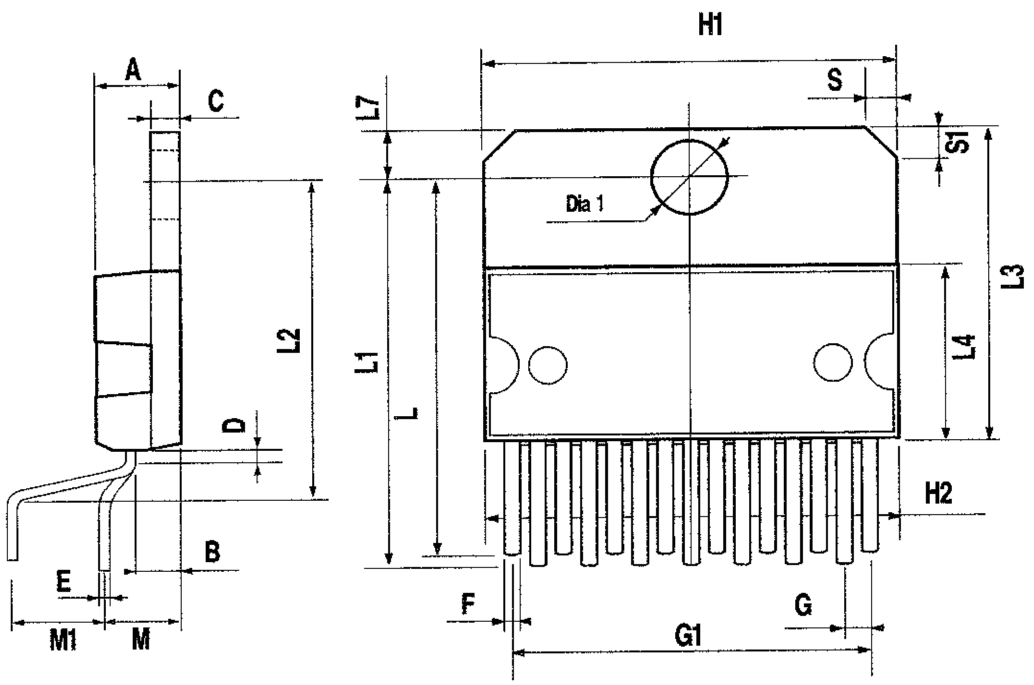
R_R and R_{sense} depend from the load current

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

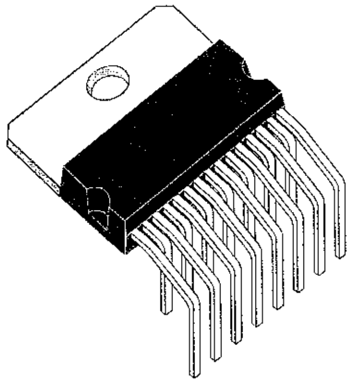


Multiwatt15 V

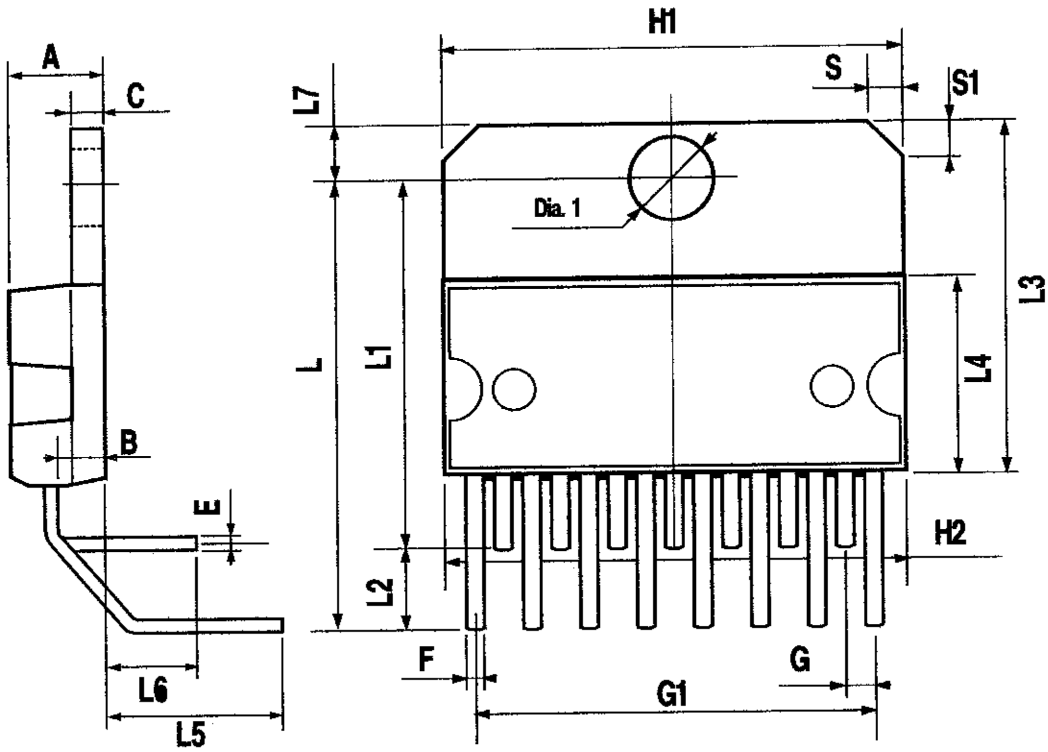


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			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



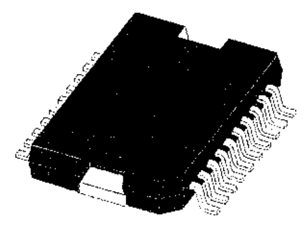
Multiwatt15 H



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	

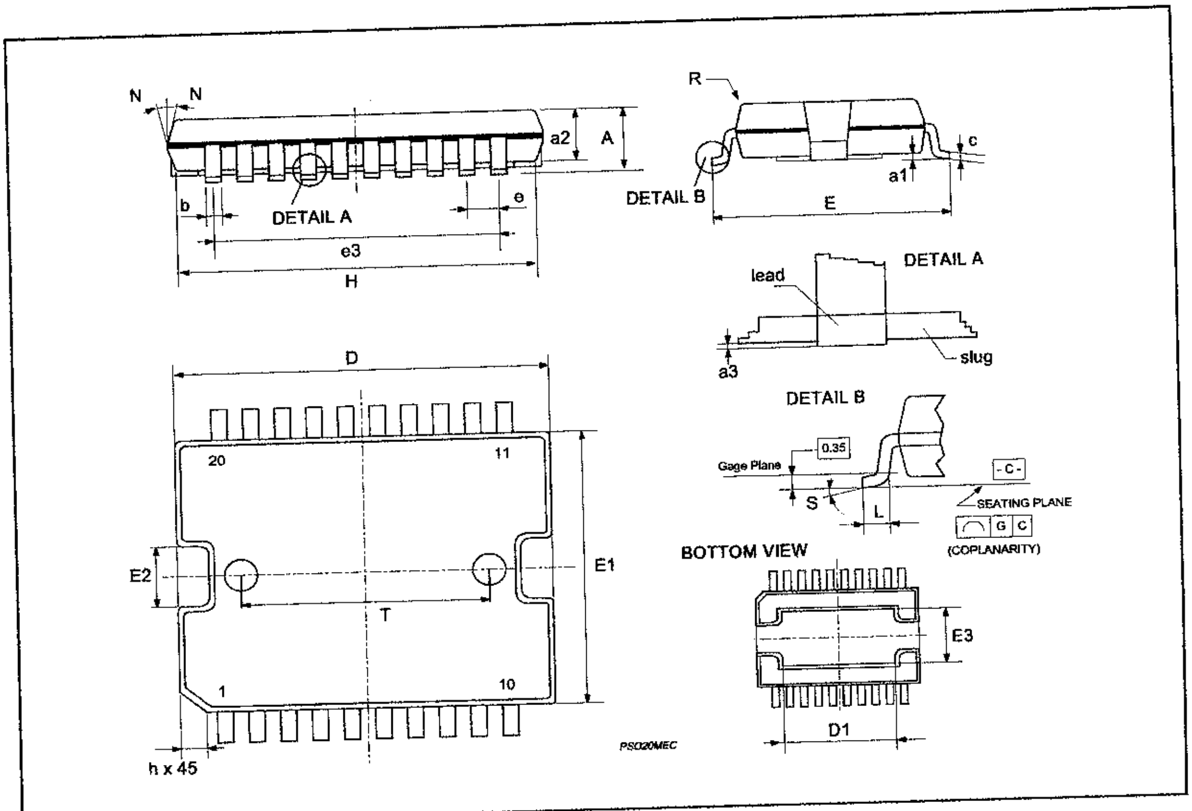
(1) "D and F" do not include mold flash or protrusions.
- Mold flash or protrusions shall not exceed 0.15 mm (0.006").
- Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20



APPENDIX G: DATASHEET FA-130RA METAL-BRUSH DC MOTOR

FA-130RA

MABUCHI MOTOR

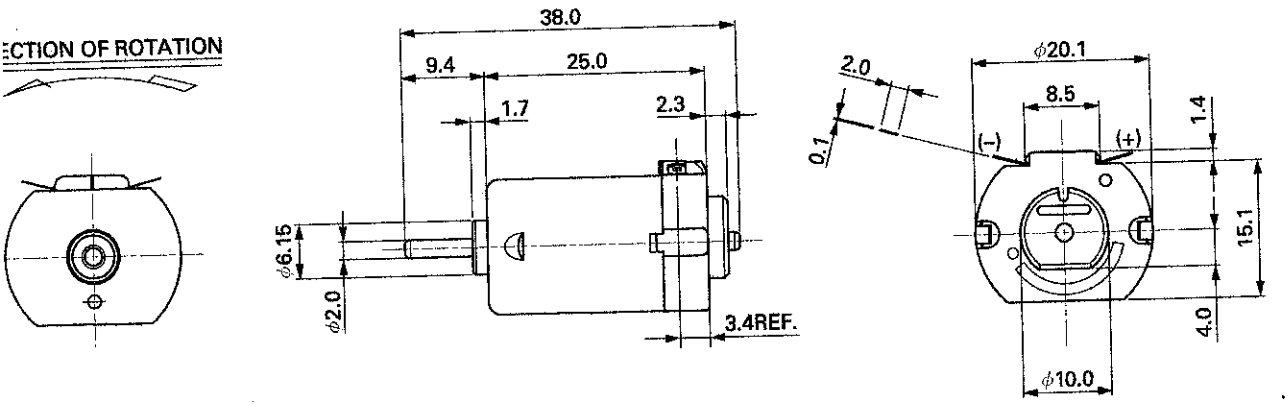
Metal-brush motors

OUTPUT : 0.2W~2.5W (APPROX)

WEIGHT : 17g (APPROX)

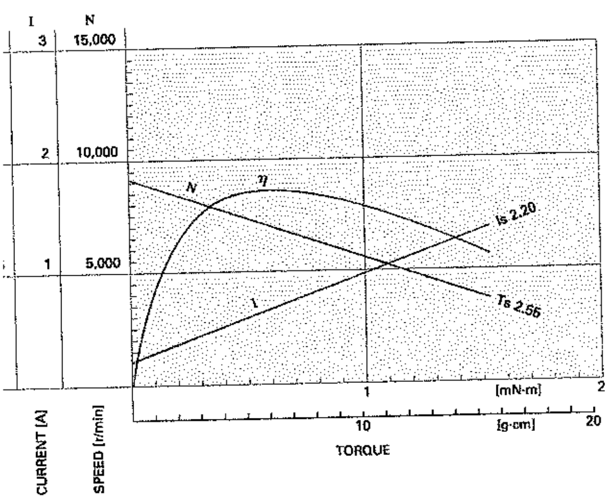
Typical Applications **Home Appliances : Hair Dressing / Beauty Appliance**
Toys and Models : Motorized Toy / Motorized Plastic Model

MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
	OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			r/min	A	r/min	A	mN·m	g·cm	W	mN·m	g·cm	A
0RA-2270	1.5~3.0	1.5V CONSTANT	9100	0.20	6990	0.66	0.59	6.0	0.43	2.55	26	2.20
0RA-18100	1.5~3.0	3V CONSTANT	12300	0.15	9710	0.56	0.74	7.6	0.76	3.53	36	2.10
0RA-14150	1.5~4.5	3V CONSTANT	8300	0.11	6150	0.31	0.55	5.6	0.35	2.11	22	0.90

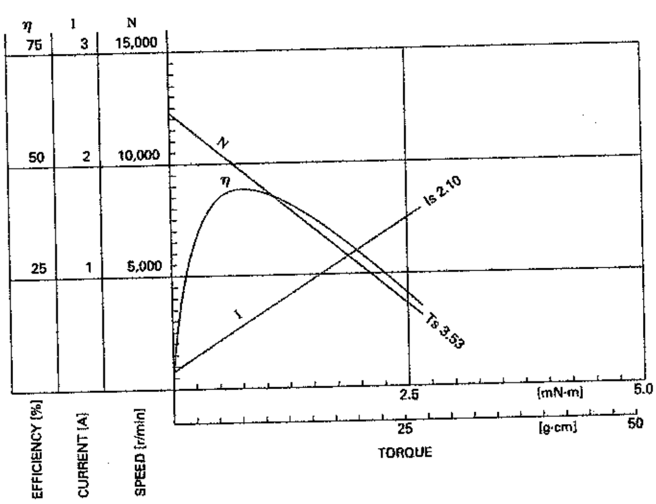


UNIT: MILLIMETERS

-130RA-2270 1.5V



FA-130RA-18100 3.0V



ABUCHI MOTOR CO., LTD. Headquarters 430 Matsuhidai, Matsudo City, Chiba, 270-2280 Japan. Tel:81-47-710-1177 Fax:81-47-710-1132 (Sales Dept.)