# AUTOMATED FOOD ATTENDANCE ROBOT

NUR IZZATI BINTI ABDULLAH

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

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER 2007

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

Signature : ______________________________

Author : NUR IZZATI BINTI ABDULLAH

Date : NOVEMBER 2007

# DEDICATION

Specially dedicated to:

- My late father, who really support me doing this project until his last breath, my mom, family, beloved ones and all friends -

## ACKNOWLEDGEMENT

In preparing the thesis, I was in contact with many people, researches, and academicians. They have contributed towards my understanding and thought. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Pn. Nor Maniha Bt Abd Ghani, for encouragement, guidance, advices and supervision. I am also thankful to my unofficial co-supervisor, En. Mohammad Fadhil Bin Abas, for his guidance and advice regarding PIC programming. Not forgetting, Encik Mohd Azlan Bin Sayuti, thank you for your motivation and guidance related to the mechanical part of the robot. Without their continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Faculty of Electrical and Electronic, Universiti Malaysia Pahang (UMP) for funding my final year project. Laboratory instructors, thank you for all your cooperation especially sacrifice your resting time during weekend to open the laboratory.

My sincere appreciation also extends to all my colleagues especially Khairul A'alam Bin Abdul Ghani and others who have provided technical assistance and support. Good luck in your future undertaking. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members for their non stop support. Last but not least, thanks God for giving me a healthy body to finish this project. Hopefully this project can contribute something useful to the committee.

## ABSTRACT

This project presents the design and development of a robot from disposal raw material, name automated food attendance robot. The function of the robot is to deliver a food to unable people especially patient in hospital and old folk's home from pantry to their bed with just pressing a start button. This robot will follow the line tracking which is black in colour. To sense the line, three sensors are used to detect right, center and left line. This project consists of two parts which are software and hardware development. The software development involved the programming for PIC 16f877 microcontroller. Melab programmer and Microcode studio are used for this purpose. Since the raw material that used is mostly aluminum, riveting process is used to build up the body of the robot. After few experiment done, the project is not fully succeeded because of the mistakes in choosing the material as will be discussed in this thesis.

# ABSTRAK

Projek ini membentangkan pembangunan serta reka bentuk sebuah robot dari bahan-bahan terpakai yg diberi nama robot penghantar makanan automatik. Fungsi utama robot ini adalah untuk menghantar makanan kepada orang-orang yang kurang upaya terutamanya pesakit di hospital dan orang-orang tua di rumah orang-orang tua bermula dari dapur ke katil pesakit.Robot ini akan mula berfungsi sebaik sahaja punat mula ditekan.Pada masa ini, kerja penghantaran makanan ini dilakukan oleh atendan di hospital manakala sukarelawan di rumah orang-orang tua. Robot ini akan menjejaki garisan berwarna hitam sebagai panduan untuk sampai ke destinasi yang dikehendaki. Untuk memastikan robot ini dapat menjejaki garisan berwarna tersebut, tiga pengesan dipsangkan untuk mengesan garisan tersebut yang terdiri daripada pengesan tengah, kiri dan kanan. Projek ini melibatkan dua bahagian iaitu pembangunan perisian serta pembangunan reka bentuk robot tersebut. Pembangunan perisian melibatkan perisian Microcode Studio serta pemprogram Melab untuk memprogramkan PIC 16F877. Disebabkan kebanyakan bahan yang digunakan adalah aluminium, meribet merupakan cara utama untuk mencantumkan bahagian-bahagian robot tersebut. Setelah menjalani ujian beberapa kali, projek ini tidak berjaya disempurnakan dengan sepenuhnya disebabkan pemilihan bahan yang tidak sempurna yang akan dijelaskan di dalam laporan ini nanti.

# TABLE OF CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

# LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

There are various methods of determining the exact position of a robot within a place. Line follower robot is one of the methods. It is the self operating robot that follows a line that drawn on the floor. Automated food attendance robot is a line follower based robot. But in this project it is designed so that the robot can function as a food attendance robot. Below is the definition of each words used for this project:

(i) Automated means, use or introduction of automatic method or equipment in place of manual labor.

(ii) Food is referred to substance taken in to maintain life growth. The meals in this project including breakfast, lunch, tea hour, dinner and supper.

(iii) Attendance will be the persons used to deliver the meals to patients.

(iv) Robot means mechanical or virtual, artificial agent. It is usually an Electromechanical system which, by its appearance or movements, conveys a sense that it has intent or of its own. The word *robot* can refer to both physical and virtual software agents, but the latter are usually referred to as bots to differentiate.

Basically, the function of the robot is to carry a tray contains of patient‘s food from kitchen to patient bed and come back to kitchen. The robot will deliver the food to unable person at hospital and residents of the old folk's home 5 times a day. It is including breakfast, lunch, tea hour, dinner and supper. The robot will follow the path as programmed in PIC microcontroller.

## 1.2 Problems Statement

Patient/old folk's/disable people can't move freely as healthier person and they need someone to serve their meal. For the time being, attendance or volunteer (helpers) in those places have doing this job. The problems occur when sometimes the helpers is not enough to do this job while attendance is doing their others job. To overcome this problem, I suggest to built the robot that can help both helpers and the patient/old folk's/disable people doing this job since robot has took place a lot of man's work for the past years.

## 1.3 Objectives

### 1.3.1 To design a robot that can carry the patient/disable/old folk's meal at their place

One of the purposes of this project is to design a robot that can carry the patient/disable/old folk's meal at their place. As stated before, this group of people can be label as disable where they need some external helps in doing their daily life works including serving their meals.

### 1.3.2 To control the robot by using PIC microcontroller due to path tracking

In order to make sure the robot is functioning, a controller to control the motor (tire) movement automatically is needed. As PIC is one of the latest technology that used to control the movement (by controlling the pulse width modulation (PWM) according to input of sensor), the decision to choose PIC microcontroller as a controller of the robot has been made. Due to a lot of pass by in both locations, path tracking is used where there is a line that connects from the place the kitchen to the patient/old folk's. The line is also used to acknowledge people that it is the path for robot. So that people can aware their step when pass thru the line.

## 1.4 Scopes of project

The scopes of this project are:

(i) The attendance robot is using the line follower based concept that is redesign to be function as carrier robot.

(ii) The robot took place in hospital/old folk's house

(iii) The robot can carry a tray at one time

(iv) The height of the robot is in range of 80cm-100cm

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Carrier Robot "CARREY"

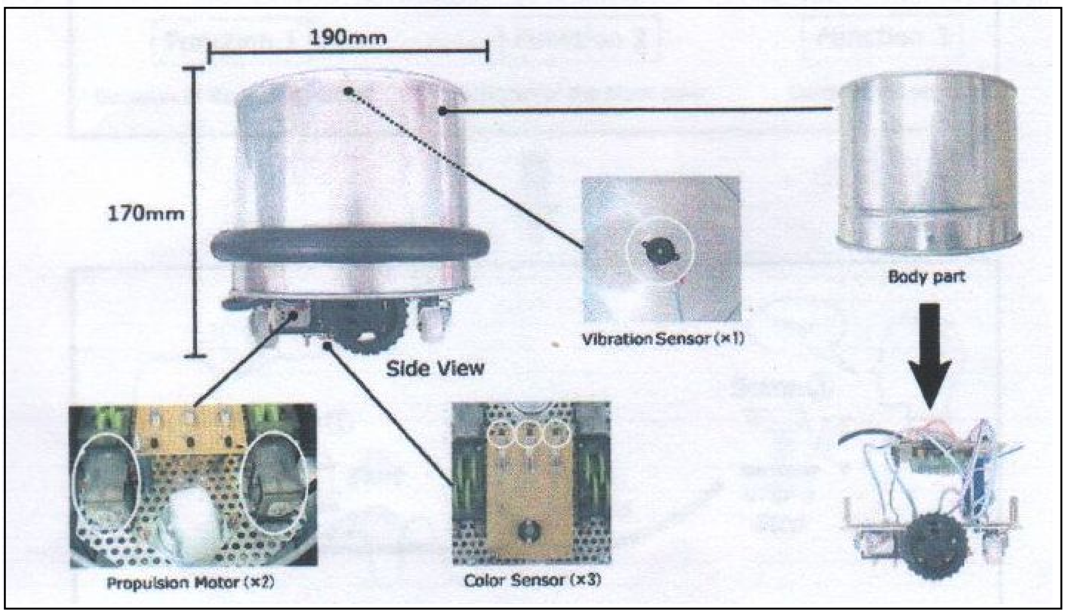



**Figure 2.1** Carrier robot "CARREY "

### 2.1.1 Robot Operation

Carrey (Figure 2.1) is a carrier robot done by Keisuke Tomio, Hiroya Igarashi and Akira Harada. The carrier robot designed was based on three functions which are detection of user's contact, detect black line colour and carrying the object. The robot system has been design moves by detection of user's contact with using primitive mechanism. The robot was installed with vibration sensor so that it knows user is putting something on the top if it. When the vibration sensor is activated, the robot will move to the desire place that has been guided by the black line. In the same time, the robot is carrying the things that have been put by the user. At the end of the line, the second user, which is the receiver will touch the robot again to stop it and take the things that robot has carried [1].

### 2.1.2 Robot Structure

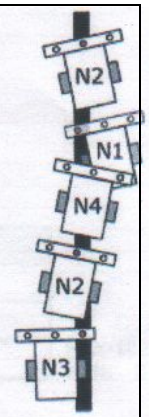
Carrey robot has a dimension of 170mm height x 190mm width. It is using microcomputer H8S/2144 to control the movement and using C language as the development language. In order to perform the function of the robot, CARREY has been installed by two sensors to give input to the robot. The sensors are vibration sensor and colour detection sensor. The vibration sensor is used to detect user contact. When vibration sensor detects the user contact, it will give input to the microcomputer to start the robot. For colour detection sensor CARREY used three infrared LED as transmitter and three photo IC. Each LED and photo IC is installed to sense right, centre and left. To move the robot, two DC motor is used. Table 2.1 shows the control table of carrier robot:

**Table 2.1** Control table of the carrier robot

| | L | C | R | Sensor States | Control States |
|---|---|---|---|---|---|
| N1 | 1 | 0 | 0 | | Turn left |
| N2 | 0 | 1 | 0 | | Move forward |
| N3 | 0 | 0 | 1 | | Turn right |
| N4 | 0 | 0 | 0 | | Move forward |
| N5 | 1 | 1 | 0 | | Turn left |
| N6 | 0 | 1 | 1 | | Turn right |

L C R

1 ··· Detection
0 ··· No Detection




### 2.1.3 Review

According to carrier robot, the robot is looking like similar to food attendance robot that I want to build in terms of the aim of the robot. Both robot is build to carrying the something. In my case, the thing is patient's food tray. In Carrey robot, the things would be office equipment. The idea is there, but the design implementation is not suitable for food attendance robot. Firstly, Carrey robot is too small to be implemented as food attendance. The height is not too height so that it needs patients needs to come down from their bed to takes the meal. Despite, because of the height is not too tall, people will not notice the appearance of the robot that might cause incident, and the food fall down. Second thing is, the robot is using the vibration to start move and stop. In cases if there is something blocking their way, automatically it will stop and the food will not be transfer to the patients.

## 2.2 Automated Food Attendance Robot

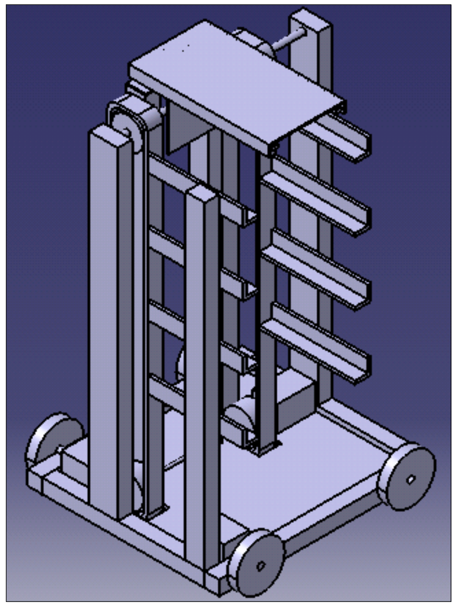


**Figure 2.2** Automated food attendance robot

### 2.2.1 Robot Operation

Figure 2.2 shows the automated food attendance robot, done by Norsinnira Zainul Azlan, Fadzilah Zainudin, Siti Farhanah Mohamad and Nazrah Abdul Aziz. This robot has been design to be as one of the service robot to deliver food to people who are unable to move. Once the button is pushed, the robot started to carry the manually loaded tray from the kitchen to patient bed. The robot traveled by following the guiding path. As the robot travel, the ultrasonic sensor that will detects if any obstacles that obstruct its navigation and the alarm will sound when it meets obstacle. After reaching at patient bed, the lifting mechanism will lift up the tray at patient bed height and the tray is pushed to the table. Then, it will continue to travel until it arrives at another bed and repeat the tray lifting and pushing process. For this prototype, the robot can carry two trays for a trip [2].

### 2.2.2 Robot Structure

The structure of the robot can be divided into three parts which are mobile platform, tray lifting mechanism and tray pushing mechanism. To ensure the mobile platform can support the entire load, the aluminum profile and plate is chosen for the construction of the robot. For robot motion, two DC motor with gearboxes is used to possess high torque in order to carry heavy load. As for brain of the robot, The Motorola MC68HC11 processor is used to control the robot to follow the line tracking. Two types of sensors are installed to the robot. One to sense the line, which is line following sensors (five sensors) while the other one is ultrasonic sensor that used to detect obstacle.

Tray lifting Mechanism is used to lift up the tray from the initial condition of the tray to the height of the patient bed. The tray holder is lift up by using PU timing belt as shown in Figure 2.3. In the other hand, tray pushing mechanism is install to make the robot can push the tray from tray holder to the patient bed or table. A rack and pinion driven by power window motor is used to accomplish this task. In this mechanism, the rotational motion of the motor is converted to the linear forward and backward motion of the tray.

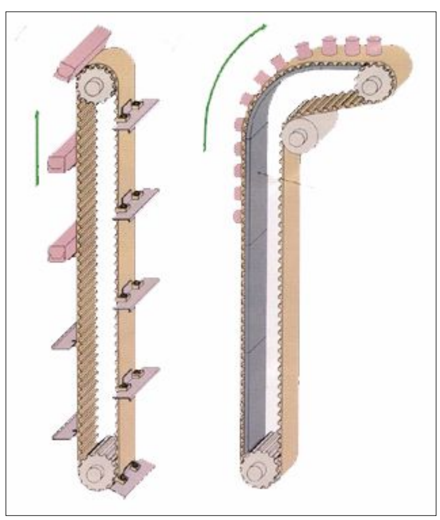


**Figure 2.3** PU timing belt

### 2.2.3 Review

According to explanation of the robot, it shows a complexity of the robot that makes the cost of the robot becomes high. In my opinion, some of the mechanism is not really necessary to include at the robot in order to reduce the cost. The terms that I referred to is the tray pushing and lifting mechanism. It is good to have those parts, but it is not the main important things to be included since it takes more than half of the total expenses for the lifting and pushing mechanism only. The another reason that it is not really necessary is, if we realized in hospital, the height of the bed or the table is varies. It depends on the situation of the patient. When the height is changed, it will effect the food in the tray is spilled, because it is not pushed at the same level. Thus, the height of the table or bed as the reference height can not been made, unless a sensor to sense the exact height of the table or bed at the current time is install. Third is, our mission to deliver the food might be accomplish. The problem is, the robot might deliver the food the bed where no patient is there because no measurement has been made to indicate there is a people or not.

As for automated food attendance robot, it just simple and more cost effective. Only a tray holder at the end of the robot to hold the tray and a limit switch is needed. The limit switch is connected at the tray holder. When the tray is still there, the load from tray will push the button and make the limit switch normally closed. Once patient takes the tray, the limit switch will be normally open. Until then only the robot will move to the next station.

# CHAPTER 3

# METHODOLOGY

## 3.1 Overview

In order to gain information for literature review to done this project, a lot of relevant and important information can be obtained via surfing the internet, browsing books and journals and also with the assistance from supervisor in charge and also person that have done similar project before.

Several methods need to be implemented in order to make sure this project success. Research is one of a good method as it can give some knowledge while doing this project. Through these researches, a lot of information gained.

During the first semester, the project is more on gaining information towards the project. This process is called as literature review and researches. It is continues with the robot, circuit, and program development in second semester. The methodology flowchart is shown as in Figure 3.1

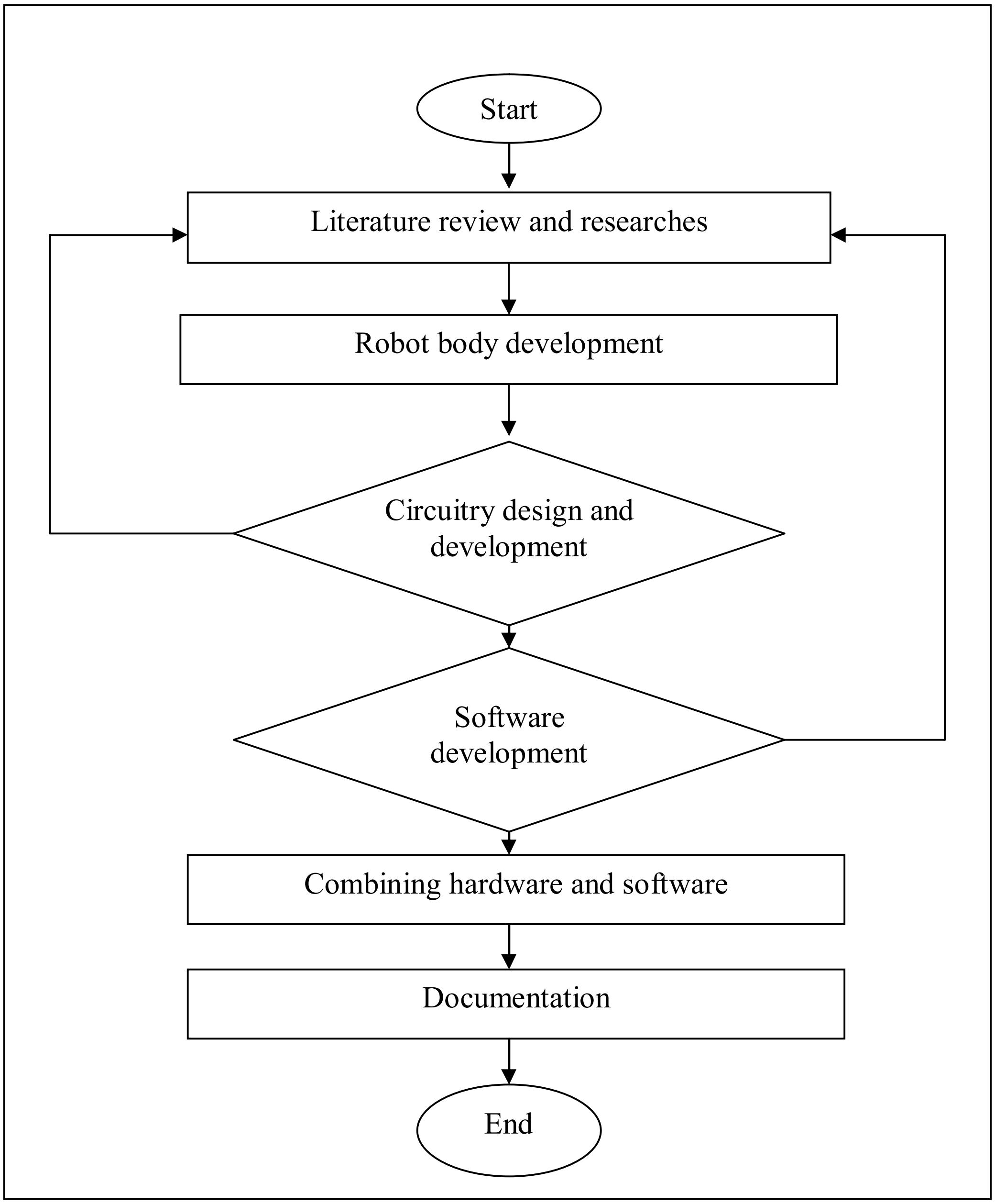



**Figure 3.1** Methodology flowchart

### 3.1.1 Literature review and research

The project begins with the literature review. In this part, all the information related to the project is gain. Before beginning to search the material, first it is need to understanding in details about the project so that the design of the robot can be drafted. Things that should be considered in this project are the place where the robot will took placed, who is the target for this project, the height of the patient bed and also weight of the meals. Rather than that, information regarding to PIC microcontroller is also needed. My referenced is based on the project that is similar to my project that has been fully succeeded. In general, this robot is line follower type robot that is redesign to be functioning as automated food attendance robot. As mention before, most of the information obtained is from internet and books source. Despite, some observation has been done to gain information related to the height of the patient bed and also weight of the meals of patient.

### 3.1.2 Robot body development

After obtaining the information in literature review section, the decision to start the project with the body development has been made. Before start the development, the design of the robot was sketched and the material that will be used to build the robot was estimated. The body development of this robot takes about three weeks. Most of the part of the body comes from used and damaged trolley bag. Than, it is redesign to becomes food attendance robot that follows the specification that I need.

### 3.1.3 Circuitry design and development

Next, after the development of the body design is finished, the project continues with the circuit design and development. The circuit design includes the circuit design for the sensor, the PIC microcontroller connection an also the motor driver connection. Circuit design starts from a single sensor to three sensors, than followed by PIC microcontroller connection and the motor driver connection. A step by step progress should be done for troubleshooting purpose. While designing those circuits, a lot of research has been done. Basically those circuits are provided in the internet and books. Some adjustment must be done with the circuitry in order to make sure the objectives and scopes of this project are followed. After the circuit is functioning, the circuit is soldered on the strip board. When I face the problems regarding the circuit, I need to review back to know what is the caused of the problem.

### 3.1.4 Software development

The robot can not be functioning without the appearance of the program. PIC microcontroller is the brain of the system of the robot. Micro Code Studio program is used to write the program and to check error of the program while Melab Programmer is used to program the PIC 16F877.A step by step process should be done in order to make easier to troubleshoot the error. The first step is to make sure that the line detecting sensor is functioning. For this purpose, three LED is connected between the output of the sensor and input PIC that indicates which sensor is functioning, right, center, or left. Second step is to check the analog to digital (ADC) program is functioning. This can be done by making simple program for ADC. For my project, I used program to ON and OFF LED depends on which sensor is detecting the black line. The third step is to change the ON and OFF LED program into ON and OFF motor. This has been done part by part according to truth table. For example, when the right sensor sense the track, the left motor will rotate while right motor stop. This shows that the robot is turning to the right. After that, proceed to the other side of sensor. Once again, the literature review is review back when problem occur.

### 3.1.5 Combining hardware and software

The next stage is to combine the hardware and software. This process is the measurement to indicate the project is succeeded or not. Sometimes, when combining the software and hardware, the robot can't function as desired. The failure may come from hardware problems or the hardware itself. To overcome the problem, the cause of the problem possibility is guess and the problem is fixed by using 'try and error' method.

### 3.1.6 Documentation

All the work through out this project needs to be documented in the last step.

# CHAPTER 4

# HARDWARE DESIGN

In order to complete building the robot, there are a lot of robot parts that should be considered in designing the robot. The parts can be divided into some components which are:

## 4.1 Components

In this section, the details about the component decision will be discussed. The components that used in projects are motor, batteries, PIC microcontroller, motor driver, sensor and tray.

.

### 4.1.1 Motor

In general, there are three types of motors commonly used in robotic and related application. They are DC motor, stepper motor and servo motor.

DC motors are inexpensive, small and powerful motors that used widely. It needs gear train reduction to reduce the speed and increase the torque output of the motor. DC motor also excellent in powering the drive wheels of a mobile robot as well as powering other mechanical assemblies.

Stepper motor in other hands is a motor that commonly used in linear performing mechanism. Usually it is used by the robotics. One of the applications is floppy and hard disk drive head motors.

The other type of commonly used motor is servo motor. A servo motors includes a built-in gear train and is capable of delivering high torques directly. Unlike DC motors and stepper motors, the servo motor has output shaft that does not rotate freely [3].

This robot needs a high torque motor. According types of commonly used motor in the market, the best motor to be used is servo motor. The common used of servo motor is the power window motor as in Figure 4.1. Despite that, the speed of the motor is not necessary for this motor since it is not used to participate in competition. If it is too fast, sometimes people can not avoid it and can make an accident between human and the robot.

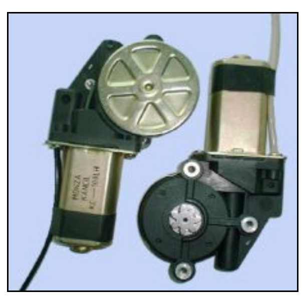


**Figure 4.1** Power window motor

### 4.1.2 Batteries (Power supply)

The most widely used of battery is rechargeable Lead Acid batteries as in Figure 4.2. It is because of easiness and inexpensive to manufacture. Even though it has been available since 1950s, it is still being used because of it is cheaper and available on the shelf. The serious problem of this kind of batteries it is large and heavy and need to always be kept charged.

**Figure 4.2** Lead acid batteries

There are three main types of lead acid batteries in market. Namely wet cell (flooded) Gel Cell and Absorbed Glass Mat (AGM) .The Gel Cell and the AGM battery is cost twice than wet cell. However, they store very well and do not tend to sulfate or degrade as easily or as easily as wet cell. The possibility of a hydrogen explosion or corrosion when using the batteries is very little. In most cases, it is believed that the AGM batteries will give greater life span and greater life cycle than a wet cell battery [4].

Despite the battery can deliver power to motor, most likely 12V to move power windows motor, the other important characteristic of the battery that I want to choose to suit my robot are:

(i) The battery must be cost wisely
(ii) The battery is rechargeable and less recharging time
(iii) The battery has greater life span

(iv) The battery can be used frequently in a day.

Based on characteristic above, I have chosen to select AGM battery is chosen to be the main supply to my robot. The Lead Acid Batteries problem, which is large and heavy, will give me an advantage. This is because the battery heaviness will give balance to the whole robot. By putting the battery at the back will balance the robot with the load of meals.

### 4.1.3 Controller (PIC microcontroller)

PIC is one of the micro controllers. PIC is used because it has its own language (high level language), name Pic Basic Pro Compiler that is simpler for the user to program it. The PIC microcontroller for this robot is PIC16F877, as shown in Figure 4.3.

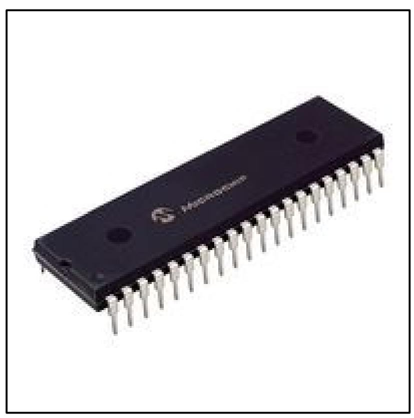


**Figure 4.3** PIC 16f877 microcontroller

The decision of this microcontroller is made because it has a feature to generate PWM. It is also have two ports for this feature. According to this feature, it suits my need where I want to control the PWM of two motors so that the robot can follow the path tracking. The datasheet of this PIC is attached in appendix A. Despite of PIC is more easy to used, it also quite cheap and easy to obtain than other micro controller like

Motorola 16HC 11 A that learnt in embedded micro controller subject that have a lot like a same features to PIC 16F877. The cost of one PIC 16F877 around RM 20 in other hand Motorola 16HC 11A is about RM 40. The data sheet of the PIC 16F877 as in appendix A

### 4.1.4 Driver

At first, IC motor driver which is L298 was selected to be the motor driver. It has 4 inputs to control the motion of the motors and two enable inputs which are used for switching the motors on and off. In application of PWM to the IC, the PWM with variable duty cycle can be connected to the enable pins. In designing this driver to the motor, the diodes is connected to prevent back EMF of the motors from disturbing the remaining circuit.

After a long research, L298 is founded not suitable for a power window motor since power window motor required higher amp than L298 can go. As shown in Figure 4.4, the driver used is specially designed for power window motor, MD30A driver. MD30A is designed to drive high ampere DC brush motor. It is a full bridge type motor driver intended for wide range of robotics and automotive applications.

**Figure 4.4** MD30A motor driver

With minimum interface, the board is ready to be plugged and play. The board also includes two push buttons for fast test run. The driver is ready to drive high current motor by just connecting the driver with supply. The specification of the motor driver is as below:

(i) Able to provide high current up to 30Amp.
(ii) Able to control heavy duty motor such as Power Window Motor, Power Wiper Motor or even Starter Motor.
(iii) Capable to control a motor in bi-directional.
(iv) Speed control can be accomplished by Pulse Width Modulator (PWM).
(v) Input voltage for motor is determined by users (12VDC, 24VDC, etc).
(vi) Easy to control by connect 5 lead wires (Gnd, PWM, CCW, CW, Vcc) to PIC controller.

The details according to MD30A can be referred as in appendix B.

### 4.1.5 Sensor

The most important part for my robot is the sensor. Without appearance of a robot, robot will not have any input. It will be just like a machine. Robot needs sensor to let them know what is happening in their world and be able to react with the changing situations [3].

In this part, the sensor used in for detecting line was considered. The cheapest choice of diode is to use standard red LED's as for light source. The LED used is white LED since it can give more light to the detector comparing to the other colour of LED. As for the receiver, phototransistor is used. Phototransistor that used is L14G1. Instead of using standard red LED's some said that, the use infra red light is better. Infra red gives some advantages is based on there isn't so much infra red light in sunlight as there is red light and that's why it's possible to make detectors with build in daylight filter. As its pair, the infrared detector is used. But, since the robot is used in door, the consideration of the sunlight can be ignored. So, it is not important to used infrared light to replace the LED.

How the sensor does works? At first, emit the white LED towards the floor. It will reflect almost all light if the floor is white in colour. As the result, the phototransistor will detect a strong signal comes from the reflection of the LED light mean while if the floor is black, it absorbs almost all the LED. In other words, the robot can distinguish between white and black line by detecting the strength of the reflected light signal which will given different value of voltage. Below is the structure of detection of white and black.

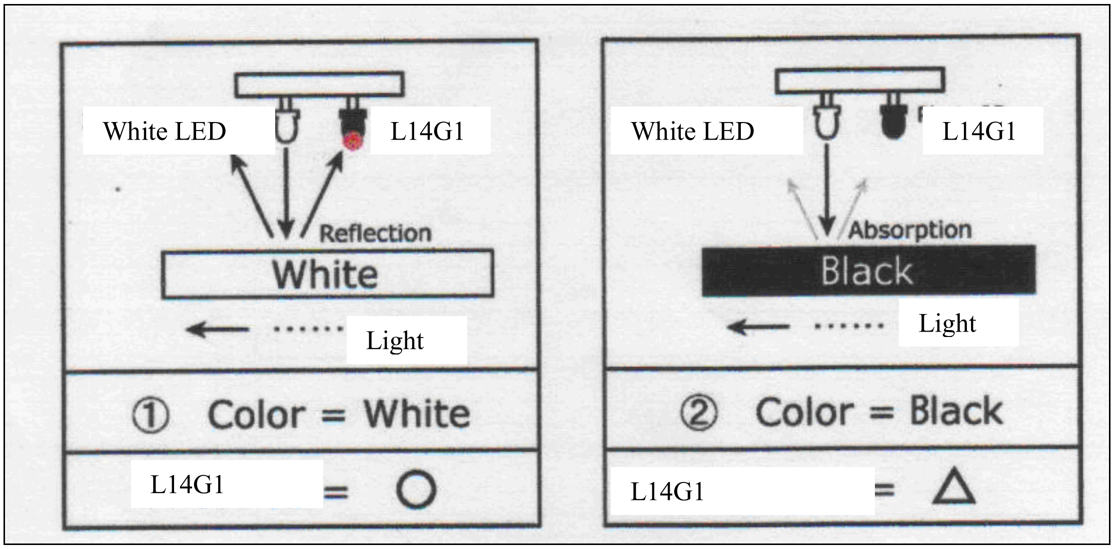



**Figure 4.5** Structure of detection of white and black

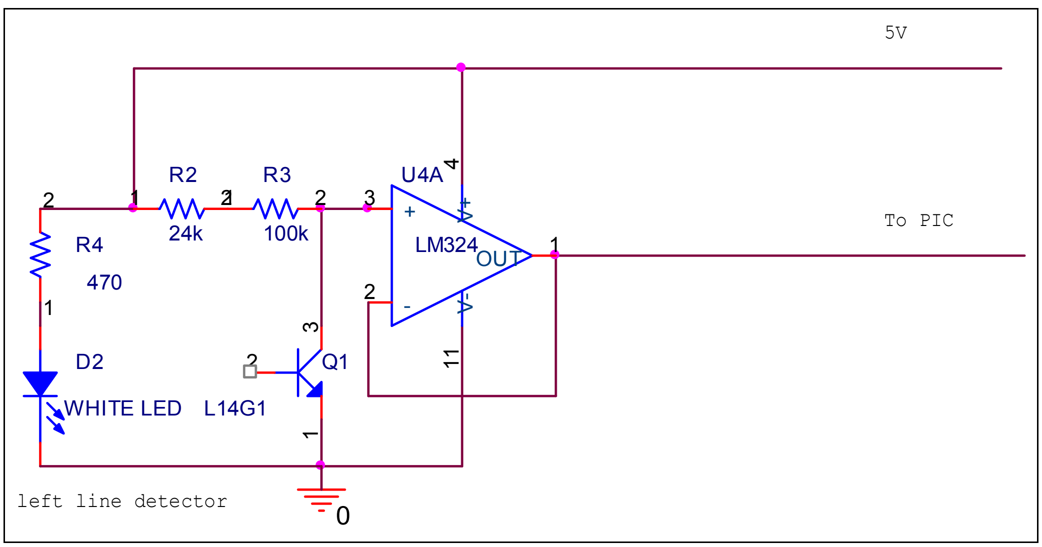



**Figure 4.6** Single line detection circuits

Figure 4.6 shows the single circuit of line detection sensor. At first, when I develop the circuit as in Figure 4.6, the difference voltage that should be when it detects line is not obtained. Since the white LED is emitting the light, prediction has been made that something wrong with phototransistor connection. So, I change the phototransistor is replaced with the LDR which only has two pins. As a result, reading of 0.73V when it detects black line while 0.24V to 0.4 V when it is white line was obtained. Since I got the result, so I predict that the connection of the phototransistor is wrong. After the pin location of the phototransistor, the voltage reading is as expected. The circuit gives a 2-3.5 V when it detects black line depends on the distance between the light sensor and the line. As for white line, it gives a measurement of 0 - 0.14V. The details regarding to phototransistor and LM324 can be referred as in appendix C and D. For the full circuit drawing, see appendix E.

### 4.1.6 Tray

Figure 4.7 shows the tray that will be used to put the patient's food. The tray that I selected is the design of the tray that always used in hospitals. Despite of it's design is frequently used, the tray is also have enough space to hold the tray which is 2cm. This tray will be the reference in designing the width of the robot. The size of the tray is 12"x 16.5".

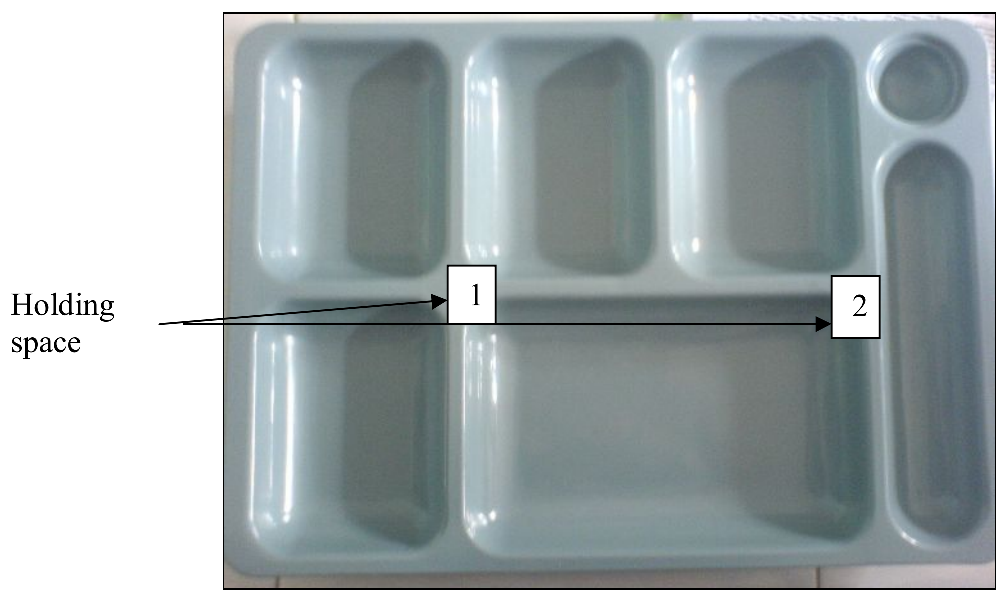



**Figure 4.7** Commonly used tray

## 4.2 Robot body

In this part the tray handle, adjustment height, sensor placement and the full version of robot body will be describe.

### 4.2.1 Tray handle

Figure 4.8 shows the design for the tray handle. The handle of the handle is sprayed with red colour so that people will notice the appearance of the robot. It has a dimension of 24 cm x 42 cm. Basically, the size is measured according to the tray as in Figure 4.7. As we can see in Figure 4.8, the labeled 1 placed at the figure shows that it will support the side labeled as 1 in figure 4.7. The same things happen for labeled 2. By doing this design, the tray will be support by the tray handle and will not slip. A limit switch is installed at the tray handle to indicate the existence of tray.

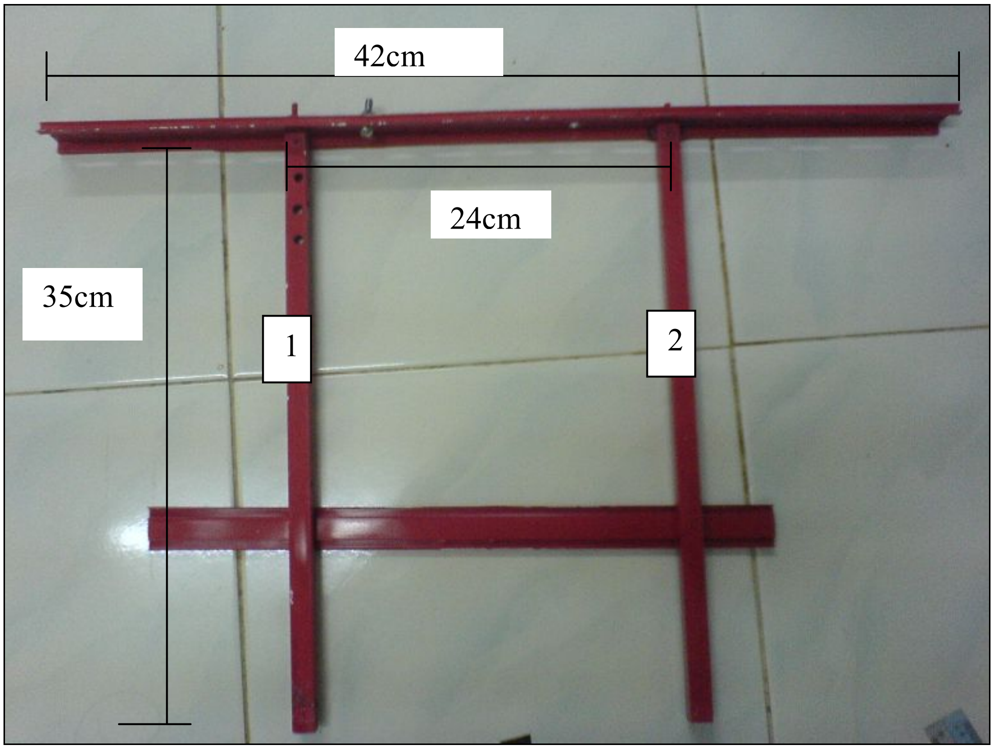
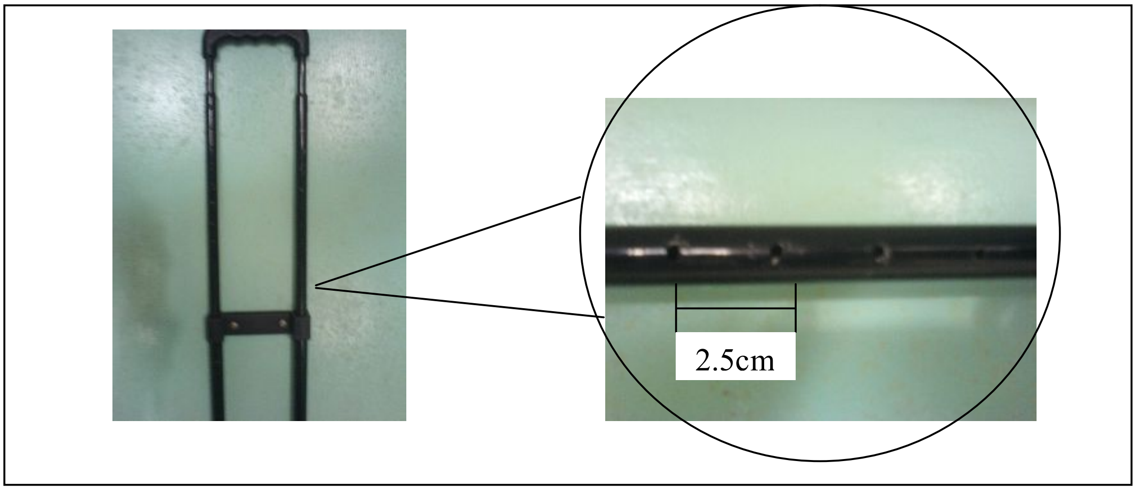



**Figure 4.8** Tray holder

### 4.2.2 Height adjustment

Figure 4.9 shows the adjuster of the robot height. The holder of the tray will be installed at the end of the adjuster. The height of the robot can be adjust from 80-100cm with the distance for each hole is 2.5 cm when it is install at the body of the robot. This means, that the height of the robot have 8 level of height with difference 2.5cm. This adjuster is made from the aluminum and consists of three different diameter of aluminum.




(a) (b)

**Figure 4.9** (a) height adjuster (b) hole to adjust the height of the robot

### 4.2.3 Sensor placement

Sensor placement can be categorized as one of the most important parts in building the robot. The sensor should be placed at the in front and at the bottom of the robot since the steering types that used is differential type drive. The steering type will be discussed on next section 4.2.4. If the sensor is put at the back, the motor will receive wrong movement since the driven is at the front. Figure 4.10 shows the placement of the sensor. In Figure 4.11, the arrangement of the sensor is shown. The distance between two sensors is to be 1cm.

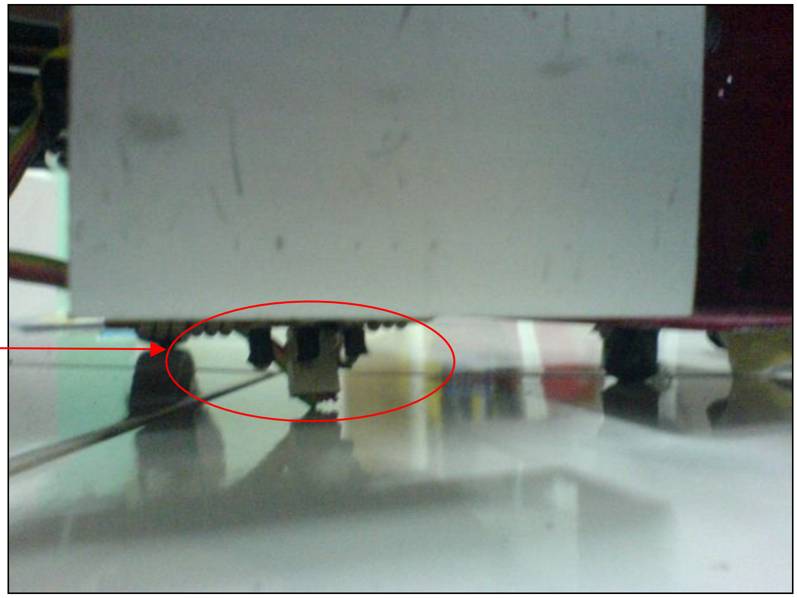



**Figure 4.10** Sensor placement from front view

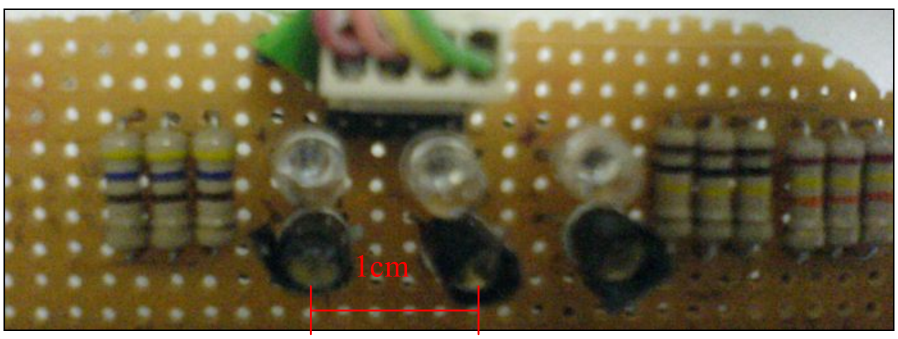
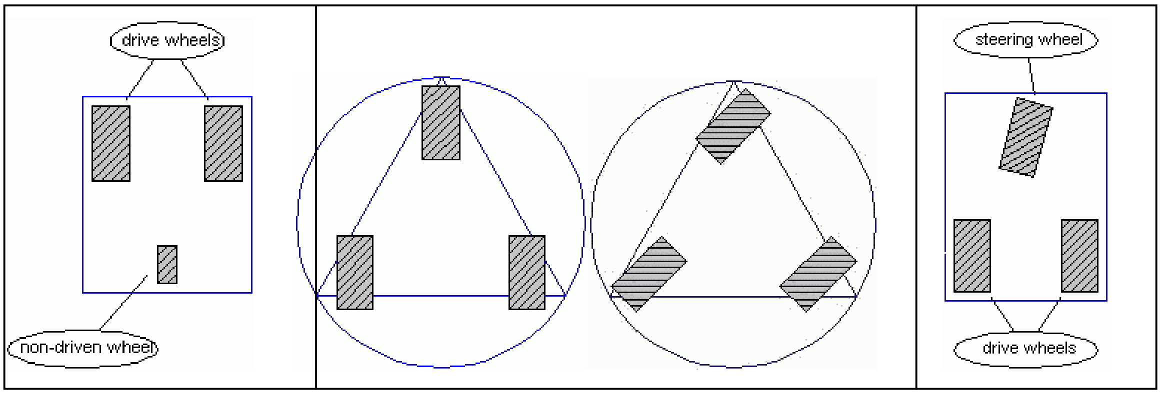



**Figure 4.11** Sensors arrangement

### 4.2.4 Steering and wheel

Basically, in this robot design, I two motor is installed. That means there will be two wheels to be control. Here are the several types of wheel driving system that is commonly used and preferred in building my robot. The types are differential drive, synchro drive and car type drive.




(a) (b) (c)

**Figure 4.12** (a) Differential drive (b) Synchro drive (c) Car type drive

**Table 4.1** Comparison of three types of wheel driving system

| Type | Differential drive | Synchro drive | Car type drive |
|---|---|---|---|
| Motor | Two- One for each drive wheel | Two- One to rotate all the wheels, and one to turn all the wheels | Two - One for translation and one for rotating the turning wheel(s). |
| Pros | Simplicity -very simple | Control- Separate motors for translation and rotation | Simplicity - One of the simplest locomotion systems to implement with one caveat |
| Cons | Control- can be difficult move in straight line | Complexity – mechanism which permits all wheels to be driven by one motor and turned by another motor | Planning - Planning is difficult because the system is non-holonomic |

Due to simplicity, the differential drive is selected while car types drive as a second choice. The con of the differential drive is to control the movement of the straight line of the robot. To overcome this problem, the motor RPM must be adjusted very often and it might require interrupt-based software. As to overcome these problems, the features in PIC 16F877 which provide both PWM and ADC features will be used.

### 4.2.5 Full version of robot body

After finishing doing the part by part of the body, the body of the robot is ready to be used. The full version of the robot is as can be seen in Figure 4.13. The drawing of the robot and it's dimension is as in appendix F

**Figure 4.14** Side view of full version of robot

# CHAPTER 5

# SOFTWARE

This chapter will discuss the step by step software requirements used in this project.

## 5.1 Analog to digital converter (ADC) program

```
DEFINE OSC 8                    'define crystal used=8Mhz'
DEFINE ADC_BITS 8               'define 8bit ADC'
DEFINE ADC_SAMPLEUS 10          'set sampling time in microsecond'
ADC VAR BYTE                    'create variable to store result'
TRISA=255                       'define port A as input'
TRISB=0                         'define port B as output'
ADCON1=1                        ' Set adcon1=1 to have reference voltage'
sense_limit VAR BYTE            ' create a byte to store the sensor limit'
sense_limit = 90                ' set sensor limit to 90'
input_sensor VAR PORTA.0        'input sensor is connected tp porta.0'
start:
ADCIN    0, ADC                 'ADC program start'
        IF ADC>=sense_limit THEN
           HIGH portb.0         'led is connected at portb.0 as indicator'
        ELSE
           LOW portb.0
        ENDIF
GOTO start
```

Above program shows the ADC program used for PIC. ADC program is used since the input for the PIC is in analog. So, the signal should be converting into digital first to ensure that the PIC can differentiate between the high and low voltage of the input. The first 3 line of the program is the terms that need to be define before ADC is used. The explanation of the program is in the blue colour font on the right.

In line four, ADC was defined as a name of variable where the result get in ADC will be store. In order to use the data obtain, it is needed to call the variable, which is ADC. Since ADC port in 16F877 is at port A, the output of the sensor will be connected to this port. For this robot, only 3 ports used which are AN0-AN3. Port A is bidirectional port. So, the port is defined as input port as in line 5. Number 1 in binary will indicates as input while 0 as output. 255 is in decimal which means that all eight bit of port A is set as input where it is 11111111 in binary. After having input port, output port should be defined also. For this purpose, port B as in line six is use.

ADCON1 is one of the register for ADC. ADCON1 is set to be 1 because porta.0-2 will have analog input while the porta.3 will be the reference voltage, 3V. More information related to ADCON1 can be referred as in appendix G. From the experiment done, the output voltage from the line sensor will give 1.05-3.00V when it detect the black line. It is depends on the height of the sensor towards the line surface. In order indicate situation where it detect line or white colour, sense limit is used. For this purpose, I take the minimum voltage as the sense limit. The calculation to find the sense limit is shown below:

$$\frac{\text{Sense limit}}{255 \text{ bit}} \times 3\text{V} = 1\text{V}$$

$$\begin{aligned}\text{Sense limit} &= 255 \text{ bit}/3 \\ &= 90 \text{ bit}\end{aligned}$$

The purpose of this stage is to check the ADC program is correct. So that, only one input is used to avoid the program become complex an easier to debug if problem occurs. The instruction from start up to GOTO start is a simple program used to ON the LED when sensor is detecting black line while OFF when it is white line.

## 5.2 Left sensor input

```
DEFINE ccp1_reg portc   'line 4-for output pwm purpose'
DEFINE ccp1_bit 2       'line 5-for output pwm purpose'
DEFINE ccp2_reg portc   'line 6-for output pwm purpose'
DEFINE ccp2_bit 1       'line 7-for output pwm purpose'
start:
GOSUB sensor_scan       'Go to subroutine sensor_scan'
GOTO start
sensor_scan:
        ADCIN 0,adc     '0-right sensor at channel ADC0,store reading to ADC'
        IF ADC>=SENSE_LIMIT THEN
           HPWM 0,127,5000 'o/p pwm at channel c0,50% duty cyle,5000hz freq'
           HIGH portc.4   'led is connected at portc.4 as indicator'
        ELSE
           HPWM 0,0,5000
           LOW  portc.4
        ENDIF
RETURN
```

This program step is done after the ADC program can function. It is important to do this step to see the pulse width modulation output (PWM) from the PIC when the sensor sense line. PWM is used to control the power window speed. PWM is the ratio of ON and off of the modulation.

To generate the PWM output, HPWM instruction is used. But before that, it is need to define the PWM port first. For 16F877, there are two PWM outputs which are ccp1 and ccp2. The first four lines show how to define the port. This program will generate PWM at CCP 1 when the sensor at porta.0 sense line, which is connected to the left sensor. And of course, in order to makes this program running it is needed to combine the ADC program with this program.

The result of this program can be seen by connecting the oscilloscope to the CCP1 port and LED that is connected to portc.4.The duty cycle of the output can be varies by changing the duty cycle at HPWM instruction. Duty cycle place is after the first comma in HPWM instruction. For this program, 127 show that the duty cycle is 50%. This means, to get 100% duty cycle, we need to write 255 there.

## 5.3 Right sensor input

```
DEFINE ccp1_reg portc   'line 4-for output pwm purpose'
DEFINE ccp1_bit 2       'line 5-for output pwm purpose'
DEFINE ccp2_reg portc   'line 6-for output pwm purpose'
DEFINE ccp2_bit 1       'line 7-for output pwm purpose'
start:
GOSUB sensor_scan       'Go to subroutine sensor_scan'
GOTO start
sensor_scan:
        ADCIN 2,adc     '0-right sensor at channel ADC0,store reading to ADC'
        IF ADC>=SENSE_LIMIT THEN
           HPWM 1,127,5000 'o/p pwm at channel c0,50% duty cyle,5000hz freq'
           HIGH portc.4  'led is connected at portc.4 as indicator'
        ELSE
           HPWM 1,0,5000
           LOW  portc.4
        ENDIF
RETURN
```

If look back to the left sensor input program, the program is just same. Purpose of doing this step is to ensure that the port connected to the right sensor can interface with the PIC and can generate the PWM output for another one channel since it has not been testing before. The terms that we change here is the ADC channel, where it is 0 before comes to 2 and HPWM channel from 0 to1. Just like before, this program also needed to be combined with the ADC program.

## 5.4 Center sensor input

```
DEFINE ccp1_reg portc   'line 4-for output pwm purpose'
DEFINE ccp1_bit 2       'line 5-for output pwm purpose'
DEFINE ccp2_reg portc   'line 6-for output pwm purpose'
DEFINE ccp2_bit 1       'line 7-for output pwm purpose'
start:
GOSUB sensor_scan       'Go to subroutine sensor_scan'
GOTO start
sensor_scan:
        ADCIN 1,adc     '0-right sensor at channel ADC0,store reading to ADC'
        IF ADC>=SENSE_LIMIT THEN
           HPWM 1,127,5000 'o/p pwm at channel c0,50% duty cyle,5000hz freq'
           HPWM 0,127,5000 'o/p pwm at channel c0,50% duty cyle,5000hz freq'
           HIGH portc.4  'led is connected at portc.4 as indicator'
        ELSE
           HPWM 1,0,5000
           HPWM 0,0,5000 'o/p pwm at channel c0,50% duty cyle,5000hz freq'
           LOW  portc.4
        ENDIF
RETURN
```

This program is the combination of two PWM output using one sensor. In theory, the left and right motor should have same speed to move straight. This is the relevant of doing this program. From this program, it shows that the robot will move forward when the center sensor sense line. However, there is some more instruction need to add to make the motor move. This is due to the needs to give input to the motor driver also.

All the program shown in this chapter is then will be the sub routine program and will combine with the main program. The full version of the program is as shown in appendix H.

# CHAPTER 6

# SYSTEM OPERATION

In this chapter, the system operation of the robot will be explained in details.

## 6.1 Delivering operation

The explanation about operation is easier by referring Figure 6.1. The robot starts functioning right after the start button is push. Before it starts to sense the track, it will wait until the limit sensor gives the high input. This indicates that the tray have been placed by attendance. After that, motor starts to scan sensor and start to move following the line. At the end of the line, the motor will rotate 180 degree and stop. This is because before this, the food is not facing the patient for the safety precaution. So, to make it serve to the patient, the robot is rotate $180^{o}$.

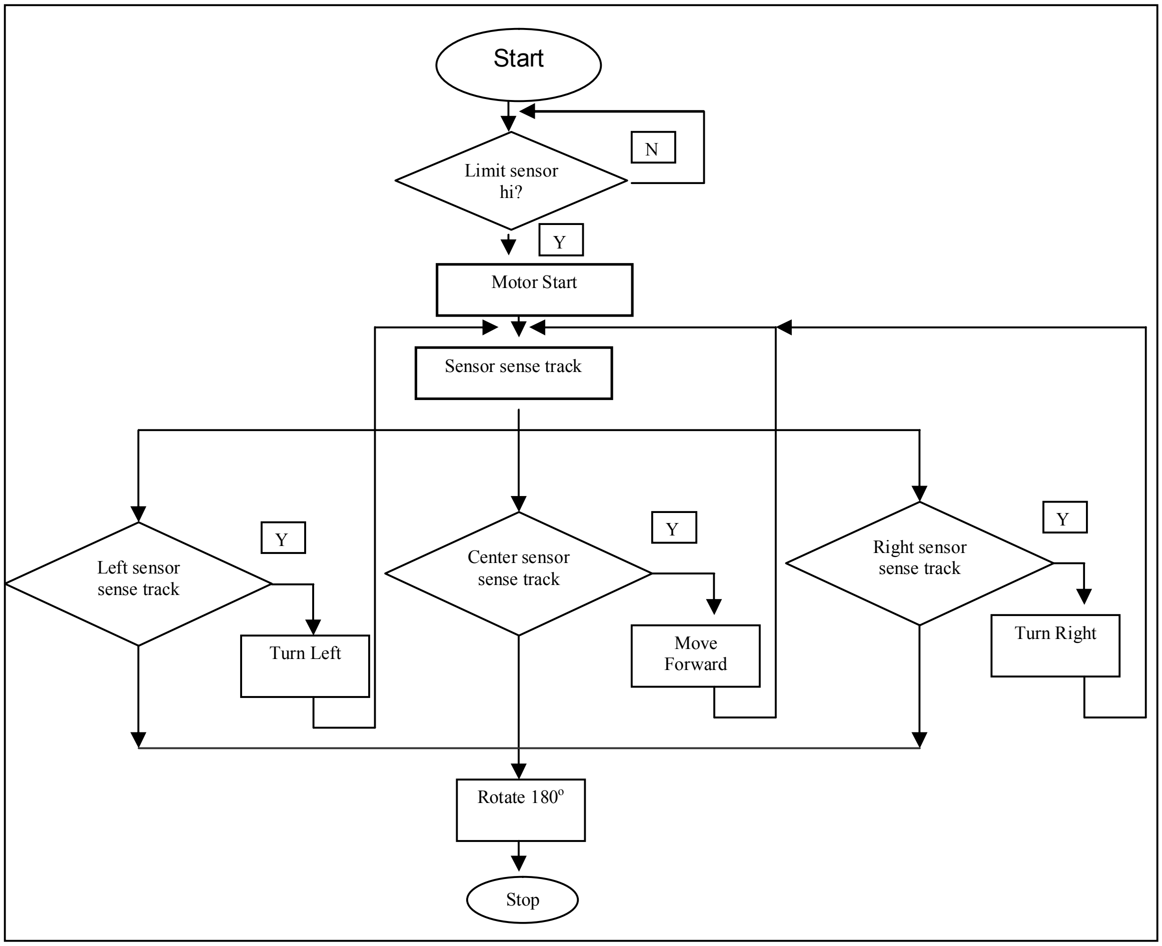



**Figure 6.1** Delivering operation flowchart

## 6.2 Going back operation

Once the robot has delivered the food, it'll be turning back to the kitchen. The operation of going back starts right after the robot rotate 180°. The robot will wait until the limit sensor gives input low. This will indicate that the patient has taken the meals. Then, the robot starts to sense the line again until the last point which is back to the kitchen. The flow chart of the going back operation is as shown in Figure 6.2

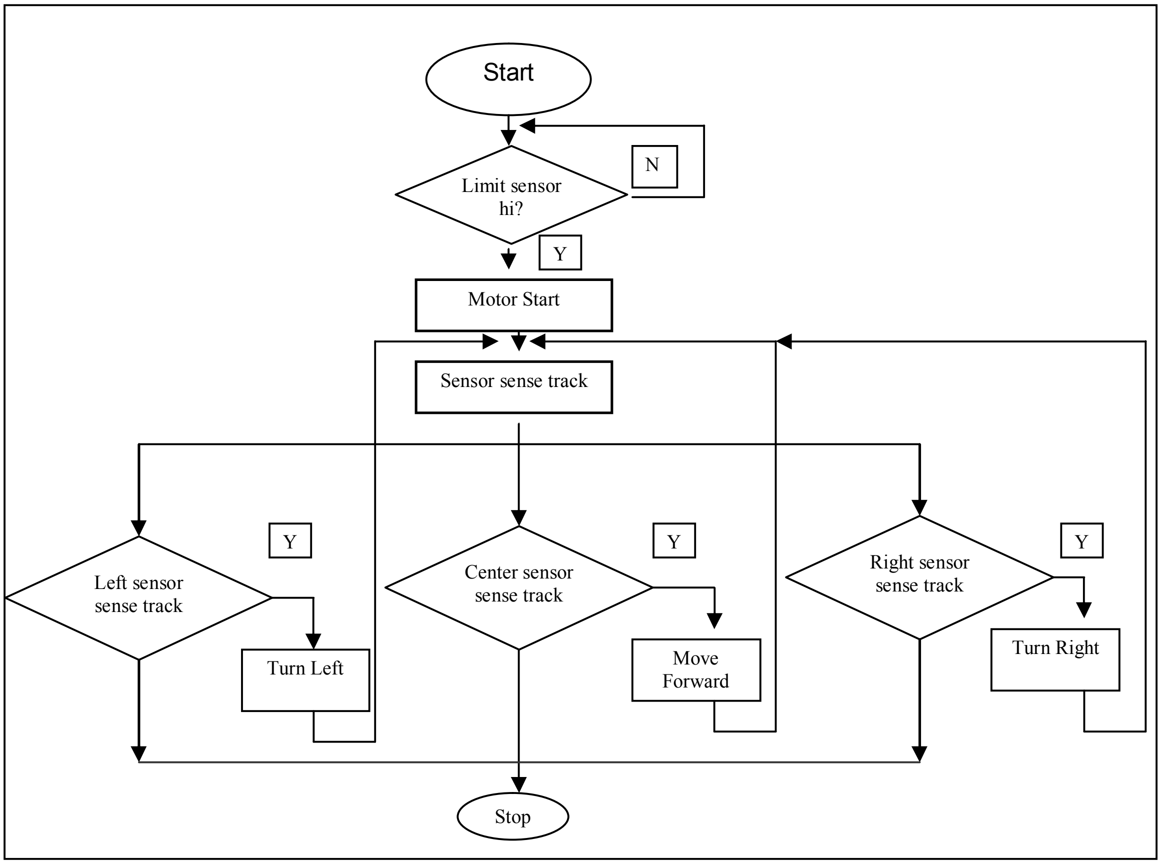



**Figure 6.2** Going back operation flowchart

# CHAPTER 7

# RESULT AND DISCUSSION

## 7.1 Result

At the end of the project, I can say that 75% of the objective and scopes is achieved. This is because the objective to used PIC microcontroller is not fully achieved. I only managed to control the motor based on table 7.1

**Table 7.1** Truth table of motor

| Sensor sense | Right Motor | | | | Left Motor | | | |
|---|---|---|---|---|---|---|---|---|
| | Ccw | Cw | Hpwm | Duty cycle | Ccw | Cw | Hpwm | Duty cycle |
| Right | Lo | Hi | Ccp2 | 0 | Hi | Lo | Ccp1 | 255 (100%) |
| Centre | Lo | Hi | Ccp2 | 255 (100%) | Hi | Lo | Ccp1 | 255 (100%) |
| Left | Lo | Hi | Ccp2 | 255 (100%) | Hi | Lo | Ccp1 | 0 |
| No sensor | Lo | Hi | Ccp2 | 0 | Hi | Lo | ccp1 | 0 |

## 7.2 Discussion

In theory, the robot should function well if the result as in Table 7.1 is gain. This is because by giving 100% duty cycle to both left and right motor should result the motor will move straight forward. The robot should have a sharp right turn if the left motor is driven 100% speed and 0% speed for right motor while it will turn sharp left if the left motor is 0% duty cycle and right motor is 100% duty cycle. But, the result is disappointing.

When the 100% duty cycle pulse width modulation (PWM) is given to both motor, the motor will turn to the left. It shows that the right motor have higher speed than the left motor. This is due to my mistake that used different specification of power window motor.

There are two solutions to solve this problem. One, change the hardware which is the power window motor while the other solution is try to solve it by try and error to get the best match PWM for both motor so that it will move forward. Since time consideration, try and error the program method is chosen. Unfortunately, the result obtained is a huge difference of PWM between two motor in order the robot to move straight. The result to move the robot forward is 45% duty cycle for right motor while 100% duty cycle for left motor. Due to time constraint and limitation, this project can not be fully success.

# CHAPTER 8

# CONCLUSION AND RECOMMENDATION

## 8.1 CONCLUSION

It can conclude that the importance in choosing the right material is very important in doing this project. If there is much more time and realize the mistake at earlier stage, the problems can be overcome and the objectives of this project will achieve 100%. Despite of using the power window motor, using the stepper motor is one of other good solution. This is because, the angle of the turning point of the motor is much easier to control according to path tracking.

To complete this project, it takes about 4 month. As an amateur to mechanical design, a lot of time and energy needs to do the hardware. Since PIC microcontroller never used before, it also takes a long period of time to understanding the program. Even though it does not achieve 100% of its objective, I still proud because at least this project have made something that is useful to the community and helps the disable people.

## 8.2 Recommendation

For the future improvement of the project, I have a lot of idea how does the robot can be friendlier user and become smart systems. My recommendations are to install auto avoiding obstacle, voice recording function and also to build a robot that can carry more trolley.

### 8.2.1 Auto Avoiding Obstacles

I suggest the robot is installed with the auto avoiding obstacle system. This is to ensure that the robot will avoid anything emergency that might occur along their trip. For the auto avoiding system, the robot will avoid obstacles in front of it and will trace back the line. So, it just not stops at the place where they found the obstacles. By installing this, the robot can be said as a smart robot.

### 8.2.2 Voice Recording Function

Voice recording function is installed to make psychological effects to the patient. Patients always need cares and attention from people. Some of them don’t want to eat, some are lonely. By installing this function, a voice of the people that they love can be record. Or maybe, the robot is recorded with the jokes of the day. So that, it might helps patient to smile and doesn’t feel lonely anymore.

### 8.2.3 Bigger robot

Bigger robot is needed to ensure that it can carry a lot of meals at one time. This will reduce the time to distribute all the foods to the patient. But, by making it big, it means that the program will become much difficult and complex.

### 8.2.4 Costing and Commercialization

**Table 8.1**: Costing

| No | Item | Cost/ unit (RM) | Quantity | Estimation cost (RM) |
|---|---|---|---|---|
| 1 | White LED | 0.40 | 3 | 1.20 |
| 2 | Phototransistor (L14G1) | 7.00 | 3 | 21.00 |
| 3 | Resistor 100k | 0.01 | 3 | 0.02 |
| 4 | Resistor 24k | 0.04 | 3 | 0.12 |
| 5 | Resistor 470 | 0.05 | 3 | 0.15 |
| 6 | LM324 | 2.00 | 1 | 2.00 |
| 7 | PIC 16F877 | 23.00 | 1 | 23.00 |
| 8 | 7805 | 1.00 | 1 | 1.00 |
| 9 | Capacitor 1u | 0.08 | 1 | 0.08 |
| 10 | Capacitor 100u | 0.15 | 2 | 0.30 |
| 11 | Heat sink | 0.90 | 1 | 0.90 |
| 12 | Header | 0.50 | 4 | 2.00 |
| 13 | Strip board | 3.00 | 1 | 3.00 |
| 14 | Wire wrap | 15.00 | 1 | 15.00 |
| 15 | IC Base 40 pin | 0.20 | 1 | 0.20 |
| 16 | IC Base 14 pin | 0.20 | 1 | 0.20 |
| 17 | Power window motor | 40.00 | 2 | 80.00 |
| 18 | Lead Acid Rechargeable Battery 12V | 60.00 | 1 | 60.00 |
| 19 | Motor Driver | 120.00 | 2 | 240.00 |
| 20 | Aluminum profile | 40.00 | 1 | 40.00 |
| **TOTAL ESTIMATION** | | | | **490.17** |

The potential of automated food attendance robot to be commercialized is possible. This is due to this robot is design to be flexible to be used at any kind of place. The terms that I referred to is the height of the robot can be adjust manually according suitable of the place height. For example, if it is needed to use in restaurant, the height of the robot can be adjusted to be the height of the restaurant table. Not just that, the size of the robot is also small, so that it can be easily kept even at a small place.

**REFERENCES**

[1] Keisuke Tomio, Hiroya Igarashi, Akira Harada. *Designing of a carrier robot "Carrey" at* office.Available at: http://www.idemployee.id.tue.nl/g.w.m.rauterberg/conferences/CD_doNotOpen/ADC/final_paper/467.pdf

[2] Norsinnira Zainul Azlan,Fadzillah Zainuddin, Siti Farhanah Mohamad, Nazrah Abdul Aziz (2007). Automated Food Attendance Robot. International Conference on Control and Mechatronics Engineering (CIM'07).May 26-29,2007

[3] Robotics.Engineering.-.Robot.Builders.Guide.pdf

[4] http://www.societyofrobots.com

[5] Rao(June 17), *What type of robot to choose?? Part – II* Available at http://www.roboticsindia.com/modules.php?name=News&file=article&sid=29
E.A PARR, (2003) "Programmable Controllers an Engineer's Guide"

[6] UMP Robocon Team (2007)

# APPENDIX A

## PIC 16F877 Datasheet

MICROCHIP

# PIC16F87X

## 28/40-Pin 8-Bit CMOS FLASH Microcontrollers

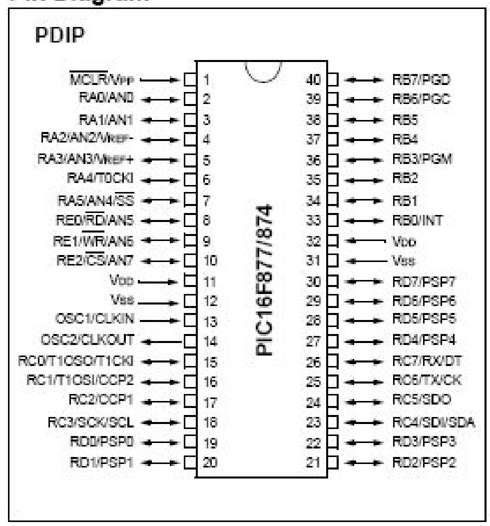
### Devices Included in this Data Sheet:

- 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 I²C™ (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)

© 2001 Microchip Technology Inc.

# PIC16F87X

### 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. |
| 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/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.


© 2001 Microchip Technology Inc.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

| Pin Name | DIP Pin# | PLCC Pin# | QFP Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | | | | | | PORTC is a bi-directional I/O port. |
| RC0/T1OSO/T1CKI | 15 | 16 | 32 | I/O | ST | RC0 can also be the Timer1 oscillator output or a Timer1 clock input. |
| RC1/T1OSI/CCP2 | 16 | 18 | 35 | I/O | ST | RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. |
| RC2/CCP1 | 17 | 19 | 36 | I/O | ST | RC2 can also be the Capture1 input/Compare1 output/PWM1 output. |
| RC3/SCK/SCL | 18 | 20 | 37 | I/O | ST | RC3 can also be the synchronous serial clock input/output for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | 23 | 25 | 42 | I/O | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | 24 | 26 | 43 | I/O | ST | RC5 can also be the SPI Data Out (SPI mode). |
| RC6/TX/CK | 25 | 27 | 44 | I/O | ST | RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. |
| RC7/RX/DT | 26 | 29 | 1 | I/O | ST | RC7 can also be the USART Asynchronous Receive or Synchronous Data. |
| | | | | | | PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus. |
| RD0/PSP0 | 19 | 21 | 38 | I/O | ST/TTL(3) | |
| RD1/PSP1 | 20 | 22 | 39 | I/O | ST/TTL(3) | |
| RD2/PSP2 | 21 | 23 | 40 | I/O | ST/TTL(3) | |
| RD3/PSP3 | 22 | 24 | 41 | I/O | ST/TTL(3) | |
| RD4/PSP4 | 27 | 30 | 2 | I/O | ST/TTL(3) | |
| RD5/PSP5 | 28 | 31 | 3 | I/O | ST/TTL(3) | |
| RD6/PSP6 | 29 | 32 | 4 | I/O | ST/TTL(3) | |
| RD7/PSP7 | 30 | 33 | 5 | I/O | ST/TTL(3) | |
| | | | | | | PORTE is a bi-directional I/O port. |
| RE0/$\overline{RD}$/AN5 | 8 | 9 | 25 | I/O | ST/TTL(3) | RE0 can also be read control for the parallel slave port, or analog input5. |
| RE1/$\overline{WR}$/AN6 | 9 | 10 | 26 | I/O | ST/TTL(3) | RE1 can also be write control for the parallel slave port, or analog input6. |
| RE2/$\overline{CS}$/AN7 | 10 | 11 | 27 | I/O | ST/TTL(3) | RE2 can also be select control for the parallel slave port, or analog input7. |
| VSS | 12,31 | 13,34 | 6,29 | P | — | Ground reference for logic and I/O pins. |
| VDD | 11,32 | 12,35 | 7,28 | P | — | Positive supply for logic and I/O pins. |
| NC | — | 1,17,28, 40 | 12,13, 33,34 | | — | These pins are not internally connected. These pins should be left unconnected. |

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.

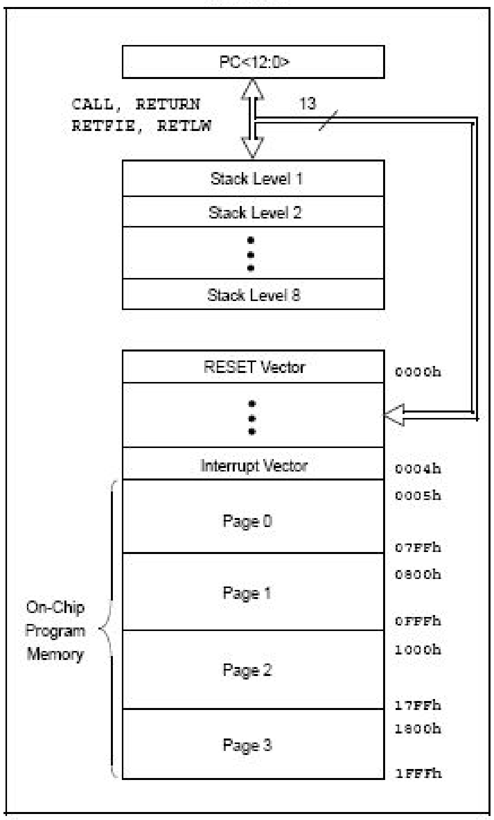
© 2001 Microchip Technology Inc.

## 2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87X MCUs. The Program Memory and Data Memory have separate buses so that concurrent access can occur and is detailed in this section. The EEPROM data memory block is detailed in Section 4.0.

Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

FIGURE 2-1: PIC16F877/876 PROGRAM MEMORY MAP AND STACK

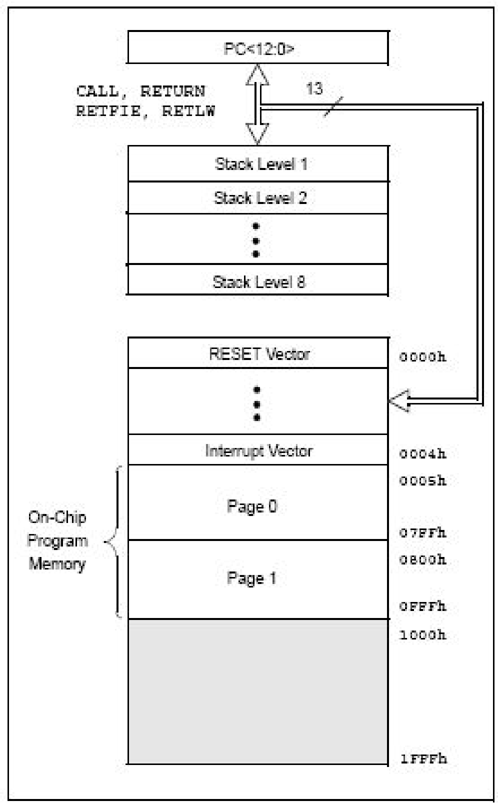



## 2.1 Program Memory Organization

The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory, and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wraparound.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-2: PIC16F874/873 PROGRAM MEMORY MAP AND STACK



# APPENDIX B

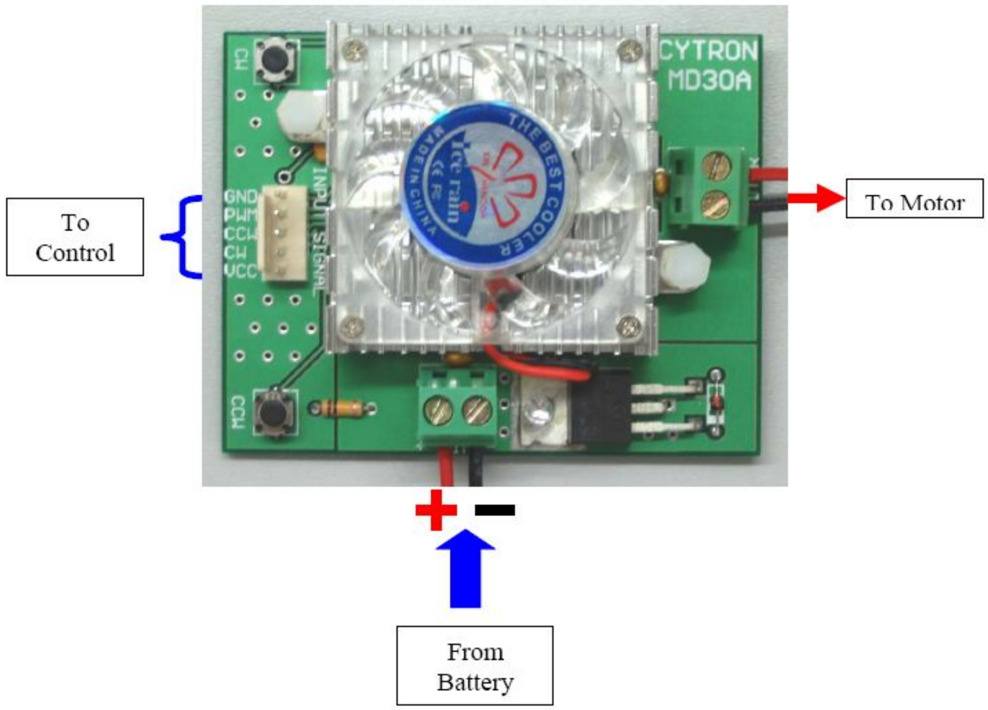
## Motor Driver Datasheet

### 1. INTRODUCTION AND OVERVIEW

MD30A is designed to drive high current brush motor or application. It is a full bridge motor driver intended for wide range of robotics and automotive applications. The board incorporates most of the components of the typical applications. With minimum interface, the board is ready to be plugged and play. It even includes 2 push buttons for fast test run. Simply add in power, this driver is ready to drive high current motor. It has been designed with capabilities and features of:

- Industrial grade PCB with heavy copper material for **high current applications**
- Each component is soldered properly and tested
- Support up to **30A maximum**
- 5V logic level compatible inputs
- 5V to 12V compatible for $V_{cc}$
- **PWM speed control** up to **10KHz**
- **Bi-directional control** for 1 motor
- Over voltage clamp
- Thermal Shut Down
- Cross-Conduction protection
- Linear current limiter
- Very low standby power consumption
- **Protection against**: Loss of ground and loss of $V_{in}$
- 2 on-board push buttons for fast testing
- Fan heat sink for fast thermal release

**2. BOARD LAYOUT**




**CAUTION**: Please ensure the battery is correctly connected to driver. One simple method to know is from on board fan. If the fan rotate when battery is connected, the connection is correct.

## 3. SPECIFICATION

Pin Function Description

| Label | Definition | Function |
|---|---|---|
| $V_{in}$* | Battery Input | Power source for motor. It can be as low as 5.5V and as high as 30V. The PCB pad is designed for terminal block. User may solder terminal block or solder the battery cable directly. |
| Motor | Motor Terminal | Terminal for motor connection. The PCB pad is designed for terminal block. User may solder terminal block or solder the battery cable directly. |
| $V_{cc}$ | Operating supply | Input for driver logic operation. The range is from 5V to 12V |
| CW | Clock Wise | Voltage controller input pin with hysteresis, CMOS compatible. These two pins control the state of the bridge in normal operation according to the truth table 4.1 (stop, brake, clockwise and counterclockwise). Both these pins are pull up internally to $V_{cc}$. Thus by default the motor is brake to $V_{in}$. |
| CCW | Counter Clock Wise | |
| PWM | Pulse Width Modulation | Voltage controlled input pin with hysteresis, CMOS compatible. Gates of Low-Side FETS get modulated by the PWM signal during their ON phase allowing speed control of the motor. |
| Gnd | Ground | Logic ground signal. Internally connected together with $V_{in}$'s ground |

* Although $V_{in}$ can support up to 30V, it is limited to 24V because the maximum voltage of on board fan is 24V. The on board fan act as $V_{in}$ indicator. If the fan turn, the polarity is correct. If the fan do not turn, please do not activate the driver, check the polarity and voltage of $V_{in}$.

Absolute Maximum Rating

| Symbol | Parameter | Value | Unit |
|---|---|---|---|
| $V_{in}$ | Motor supply voltage | 24 | V |
| $V_{cc}$ | Operating voltage | 12 | V |
| $I_{max}$ | Maximum Output Current (continuos) | 30 | A |
| $I_R$ | Reserve Output Current (continuos) | 30 | A |
| $I_{in}$ | Logic Input current (CW/CCW) | +/- 10 | mA |
| $I_{pw}$ | PWM Input Current | +/- 10 | mA |
| $T_j$ | Junction Operating Temperature | Internally Limited | °C |
| $T_c$ | Case Operating Temperature | -40 to 150 | °C |
| $T_{STG}$ | Storage Temperature | -55 to 150 | °C |

Electrical Characteristics

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{in}$* | Motor supply voltage | | 5.5 | | 24* | V |
| $V_{cc}$ | Operating supply voltage | | 5 | | 12 | V |
| f | PWM frequency | | 0 | | 10 | KHz |
| $V_{ov}$ | Overvoltage shut-down | | 36 | 43 | | V |
| $I_{LIM}$ | Current Limitation | | 30 | 45 | | A |
| $T_{TSD}$ | Thermal shut-down temperature | | 150 | 170 | 220 | °C |
| $T_{TR}$ | Thermal Reset temperature | | 135 | | | °C |
| $V_{pwl}$ | PWM low level voltage | | | | 1.5 | V |
| $I_{pwl}$ | Low level PWM pin current | Vpw = 1.5V | 1 | | | uA |
| $V_{pwh}$ | PWM high level voltage | | 3.25 | | | V |
| $I_{pwh}$ | High level PWM pin current | Vpw = 3.25V | | | 10 | uA |
| $V_{CW/CCWL}$ | CW input low level voltage | | | | 1.5 | V |
| $I_{CW/CCWL}$ | CW input low current | $V_{CW/CCW}$ = 1.5V | 1 | | | uA |
| $V_{CW/CCWH}$ | CW input high level voltage | | 3.25 | | | V |
| $I_{CW/CCWH}$ | CW input high current | $V_{CW/CCW}$ = 3.25V | | | 10 | uA |

* Although $V_{in}$ can support up to 30V, it is limited to 24V because the maximum voltage of on board fan is 24V. The on board fan act as $V_{in}$ indicator. If the fan turn, the polarity is correct. If the fan do not turn, please do not activate the driver, check the polarity and voltage of $V_{in}$.

Truth Table in Normal Operating Condition

| CW | CCW | Motor(+) | Motor(-) | Comment |
|---|---|---|---|---|
| 1 | 1 | H | H | Brake to Vin |
| 1 | 0 | H | L | Clockwise |
| 0 | 1 | L | H | Counter Clockwise |
| 0 | 0 | L | L | Brake to Gnd |

# 4. DRIVING MOTOR

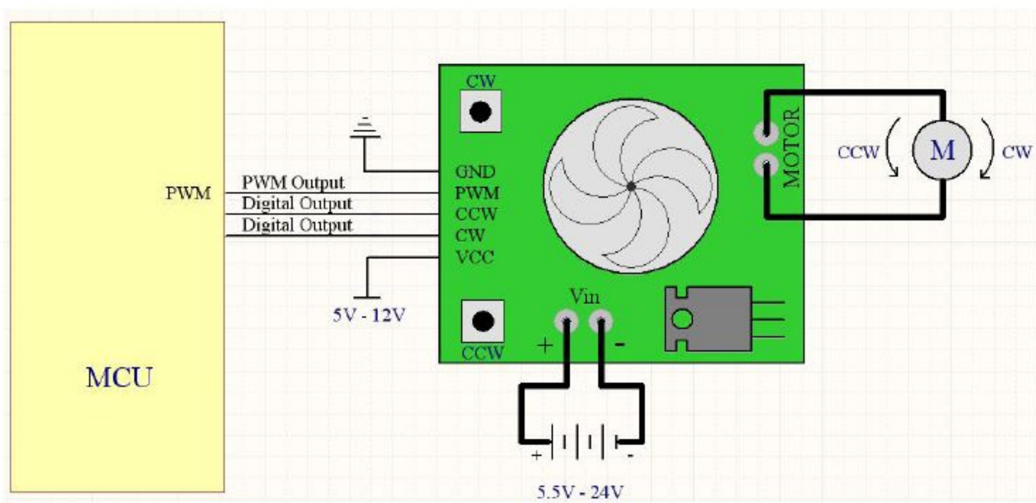
## 4.1 Connecting Battery and Motor

In a typical application, the motor power supply (battery) should be soldered directly to PCB. However, user may choose to use terminal block to lock the battery's cable. Same applied to motor terminal. The control pin come with connector and is ready for user to interface with wire. Please ensure the polarity of battery is correctly soldered/inserted to motor driver. **DO NOT** press any button or provide any signal to MD30A if the on board fan does not rotate.

5V to 12V should be supplied ($V_{cc}$) to this driver for logic operation. CW and CCW control the activation and direction of the motor, while the PWM pin turns the motor on or off for speed control. CW and CCW are internally pull-up to $V_{cc}$, thus using a switch or relay can pull these 2 pins to low stage further drive the motor. Of course, user can always use the on board switches for manual activation. As for PWM pin, user may provide a constant 5V or 12V to it if no speed control is required.

## 4.2 Connecting to Microcontroller

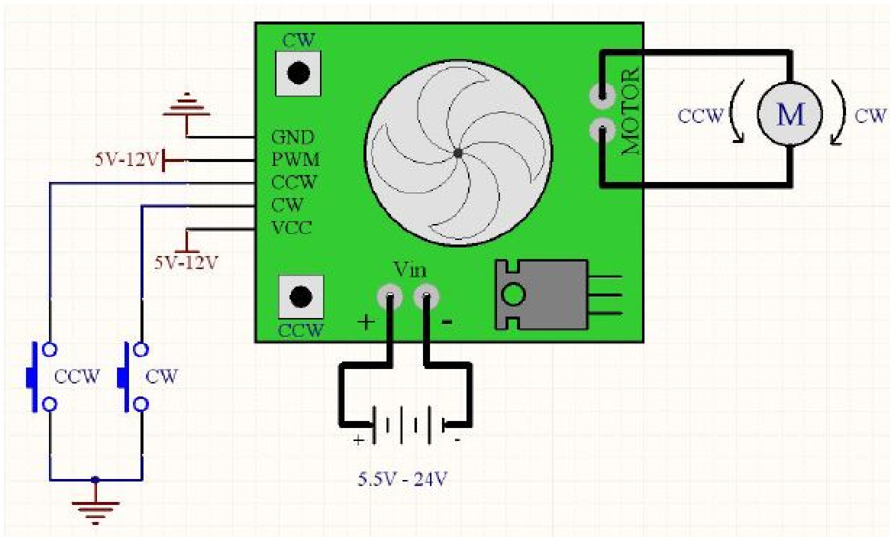
Typical Application Circuit for DC to 10KHz PWM Operation




**Note:** Please ensure the polarity of Vin is correct before pressing on button. One way to ensure is to observe the on board fan. The fan will only active if the Vin is correctly connected.

## 4.3 Connecting to switches (without microcontroller)

Typical Application Circuit using switches (no speed control)




**4.4 Real-world power dissipation considerations**

The motor drivers have maximum current ratings of 30 A continuous. However, the chips will overheat at high currents. The actual current can be delivered greatly depend on how well the fan (heat sink) keep the motor driver cool. The carrier printed circuit board and heat sink with fan are designed to draw heat out of the motor driver chips, thus do not disconnect or damage the fan. During the tests in laboratory, the result show that this driver is able to deliver short durations (about 10 seconds) of 30 A and 1 minute of 20 A without overheating. At 10 A, the driver gets just barely noticeably warm to the touch. For high-current installations, the motor and power supply wires should also be soldered directly instead of going through the supplied terminal blocks, which are rated for up to 15 A.

Many motor controllers or speed controllers can have peak current ratings that are substantially higher than the continuous current rating; this is not the case with these motor drivers, which have a 30 A continuous rating and a over-current protection that can kick in as low as 30 A (45 A typical). Therefore, the stall current of motor should not be more than 30 A. (Even if it is expected to run at a much lower average current, the motor can still draw high currents when it is starting or if low duty cycle PWM is used to keep the average current down.)

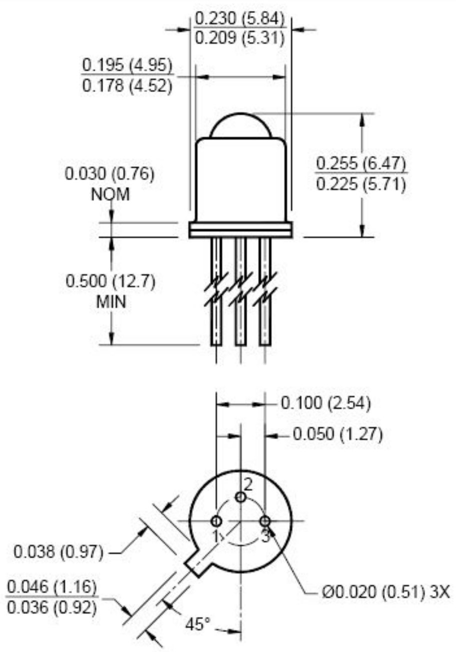
# APPENDIX C

## L14G1 Phototransistor Datasheet

# HERMETIC SILICON PHOTOTRANSISTOR

## L14G1 L14G2 L14G3

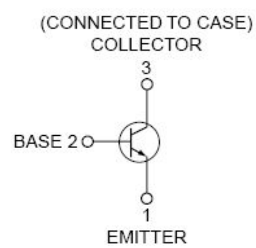
### PACKAGE DIMENSIONS




NOTES:

1. Dimensions for all drawings are in inches (mm).
2. Tolerance of ± .010 (.25) on all non-nominal dimensions unless otherwise specified.

### SCHEMATIC




## DESCRIPTION

The L14G1/L14G2/L14G3 are silicon phototransistors mounted in a narrow angle, TO-18 package.

## FEATURES

- Hermetically sealed package
- Narrow reception angle

© 2001 Fairchild Semiconductor Corporation

FAIRCHILD SEMICONDUCTOR®

# HERMETIC SILICON PHOTOTRANSISTOR

## L14G1 L14G2 L14G3

| ABSOLUTE MAXIMUM RATINGS ($T_A$ = 25°C unless otherwise specified) | | | |
|---|---|---|---|
| **Parameter** | **Symbol** | **Rating** | **Unit** |
| Operating Temperature | $T_{OPR}$ | -65 to +125 | °C |
| Storage Temperature | $T_{STG}$ | -65 to +150 | °C |
| Soldering Temperature (Iron)(3,4,5 and 6) | $T_{SOL-I}$ | 240 for 5 sec | °C |
| Soldering Temperature (Flow)(3,4 and 6) | $T_{SOL-F}$ | 260 for 10 sec | °C |
| Collector to Emitter Breakdown Voltage | $V_{CEO}$ | 45 | V |
| Collector to Base Breakdown Voltage | $V_{CBO}$ | 45 | V |
| Emitter to Base Breakdwon Voltage | $V_{EBO}$ | 5 | V |
| Power Dissipation ($T_A$ = 25°C)(1) | $P_D$ | 300 | mW |
| Power Dissipation ($T_C$ = 25°C)(2) | $P_D$ | 600 | mW |

**NOTE:**

1. Derate power dissipation linearly 3.00 mW/°C above 25°C ambient.
2. Derate power dissipation linearly 6.00 mW/°C above 25°C case.
3. RMA flux is recommended.
4. Methanol or isopropyl alcohols are recommended as cleaning agents.
5. Soldering iron tip 1/16" (1.6mm) minimum from housing.
6. As long as leads are not under any stress or spring tension.
7. Light source is a GaAs LED emitting light at a peak wavelength of 940 nm.
8. Figure 1 and figure 2 use light source of tungsten lamp at 2870°K color temperature. A GaAs source of 3.0 mW/cm$^2$ is approximately equivalent to a tungsten source, at 2870°K, of 10 mW/cm$^2$.

| ELECTRICAL / OPTICAL CHARACTERISTICS ($T_A$ =25°C) (All measurements made under pulse conditions) | | | | | | |
|---|---|---|---|---|---|---|
| PARAMETER | TEST CONDITIONS | SYMBOL | MIN | TYP | MAX | UNITS |
| Collector-Emitter Breakdown | $I_C$ = 10 mA, Ee = 0 | $BV_{CEO}$ | 45 | | — | V |
| Emitter-Base Breakdown | $I_E$ = 100 μA, Ee = 0 | $BV_{EBO}$ | 5.0 | | — | V |
| Collector-Base Breakdown | $I_C$ = 100 μA, Ee = 0 | $BV_{CBO}$ | 45 | | — | V |
| Collector-Emitter Leakage | $V_{CE}$ = 10 V, Ee = 0 | $I_{CEO}$ | — | | 100 | nA |
| Reception Angle at 1/2 Sensitivity | | θ | | ±10 | | Degrees |
| On-State Collector Current L14G1 | Ee = 0.5 mW/cm$^2$, $V_{CE}$ = 5 V(7,8) | $I_{C(ON)}$ | 1.0 | | — | mA |
| On-State Collector Current L14G2 | Ee = 0.5 mW/cm$^2$, $V_{CE}$ = 5 V(7,8) | $I_{C(ON)}$ | 0.5 | | | mA |
| On-State Collector Current L14G3 | Ee = 0.5 mW/cm$^2$, $V_{CE}$ = 5 V(7,8) | $I_{C(ON)}$ | 2.0 | | | mA |
| Turn-On Time | $I_c$ = 2 mA, $V_{cc}$ = 10 V, $R_L$ =100 Ω | $t_{on}$ | | 8 | | μs |
| Turn-Off Time | $I_c$ = 2 mA, $V_{cc}$ = 10 V, $R_L$ =100 Ω | $t_{off}$ | | 7 | | μs |
| Saturation Voltage | $I_c$ = 1.0 mA, Ee = 3.0 mW/cm$^2$(7,8) | $V_{CE(SAT)}$ | — | | 0.40 | V |

FAIRCHILD SEMICONDUCTOR®

# HERMETIC SILICON PHOTOTRANSISTOR

## L14G1 L14G2 L14G3

Figure 1. Light Current vs. Collector to Emitter Voltage

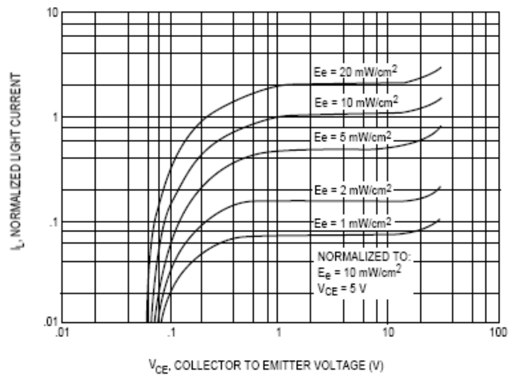



Figure 2. Light Current vs. Temperature

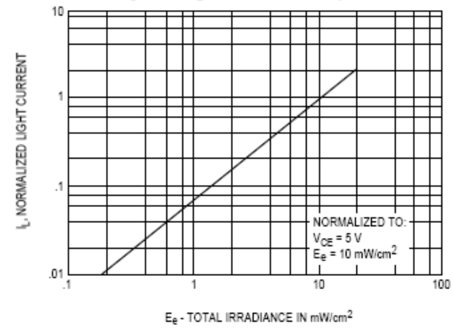



Figure 3. Normalized Light Current vs. Temperature

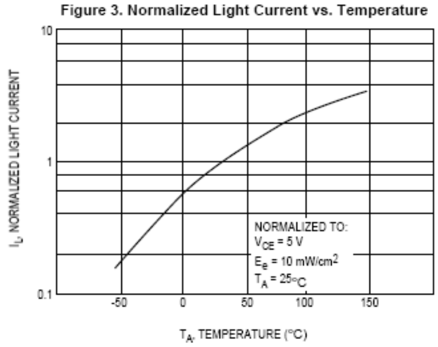



Figure 4. Switching Times vs. Output Current

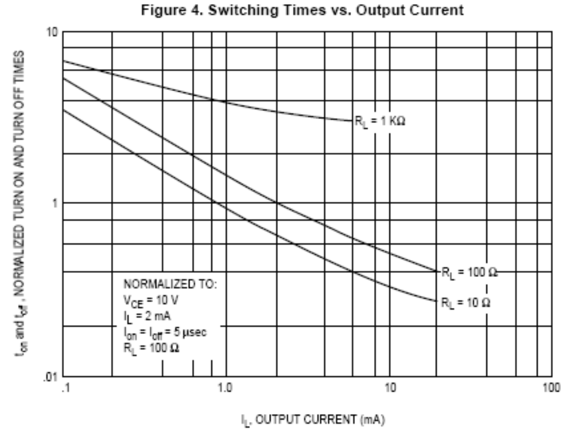



Figure 5. Dark Current and Temperature

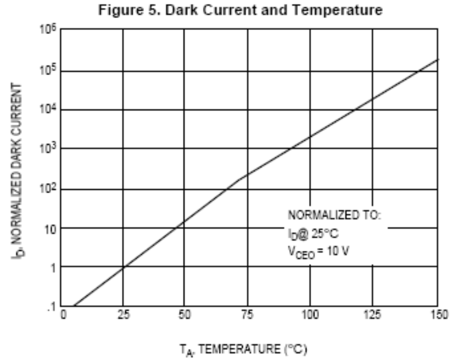
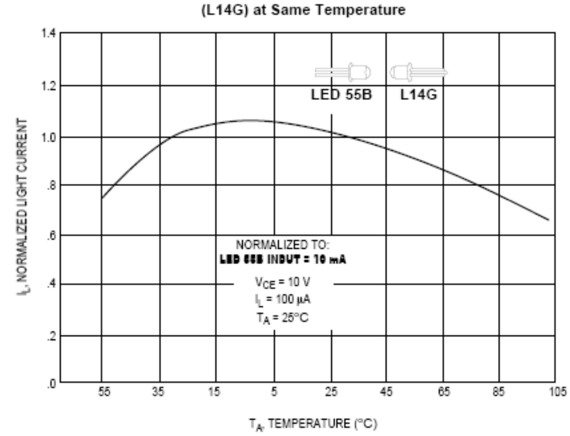



Figure 6. Normalized Light Current vs. Temperature
Both Emitter (LED 55B) and Detector
(L14G) at Same Temperature



**APPENDIX D**

**LM324 Datasheet**

FAIRCHILD SEMICONDUCTOR®

www.fairchildsemi.com

# LM2902,LM324/LM324A,LM224/ LM224A

## Quad Operational Amplifier

### Features

- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain: 100dB
- Wide Power Supply Range:
  LM224/LM224A, LM324/LM324A : 3V~32V (or ±1.5 ~ 16V)
  LM2902: 3V~26V (or ±1.5V ~ 13V)
- Input Common Mode Voltage Range Includes Ground
- Large Output Voltage Swing: 0V to $V_{CC}$ -1.5V
- Power Drain Suitable for Battery Operation

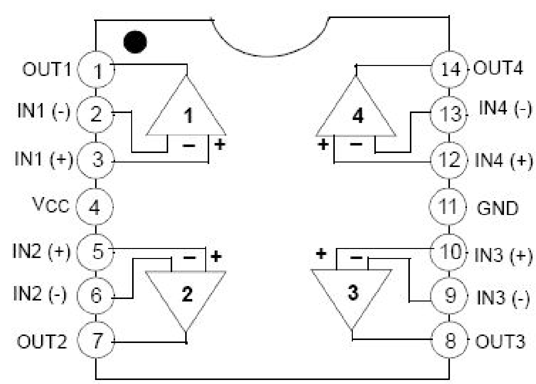
### Description

The LM324/LM324A,LM2902,LM224/LM224A consist of four independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide voltage range. operation from split power supplies is also possible so long as the difference between the two supplies is 3 volts to 32 volts. Application areas include transducer amplifier, DC gain blocks and all the conventional OP Amp circuits which now can be easily implemented in single power supply systems.

14-DIP


14-SOP


### Internal Block Diagram

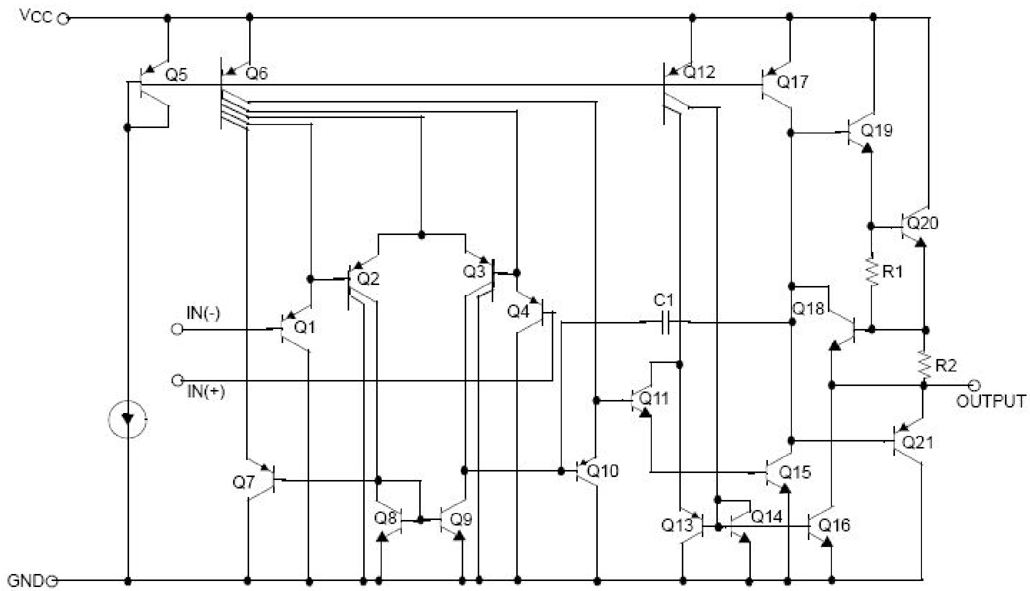



Rev. 1.0.4

©2002 Fairchild Semiconductor Corporation

## Schematic Diagram

(One Section Only)




## Absolute Maximum Ratings

| Parameter | Symbol | LM224/LM224A | LM324/LM324A | LM2902 | Unit |
|---|---|---|---|---|---|
| Power Supply Voltage | $V_{CC}$ | ±16 or 32 | ±16 or 32 | ±13 or 26 | V |
| Differential Input Voltage | $V_{I(DIFF)}$ | 32 | 32 | 26 | V |
| Input Voltage | $V_I$ | -0.3 to +32 | -0.3 to +32 | -0.3 to +26 | V |
| Output Short Circuit to GND<br>Vcc≤15V, $T_A$=25°C(one Amp) | - | Continuous | Continuous | Continuous | - |
| Power Dissipation, $T_A$=25°C<br>14-DIP<br>14-SOP | $P_D$ | <br>1310<br>640 | <br>1310<br>640 | <br>1310<br>640 | mW |
| Operating Temperature Range | $T_{OPR}$ | -25 ~ +85 | 0 ~ +70 | -40 ~ +85 | °C |
| Storage Temperature Range | $T_{STG}$ | -65 ~ +150 | -65 ~ +150 | -65 ~ +150 | °C |

## Thermal Data

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Thermal Resistance Junction-Ambient Max.<br>14-DIP<br>14-SOP | Rθja | <br>95<br>195 | °C/W |

## Electrical Characteristics

($V_{CC}$ = 5.0V, $V_{EE}$ = GND, $T_A$ = 25°C, unless otherwise specified)

| Parameter | Symbol | Conditions | | LM224 | | | LM324 | | | LM2902 | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| Input Offset Voltage | $V_{IO}$ | $V_{CM}$ = 0V to $V_{CC}$ -1.5V, $V_{O(P)}$ = 1.4V, $R_S$ = 0Ω (Note1) | | - | 1.5 | 5.0 | - | 1.5 | 7.0 | - | 1.5 | 7.0 | mV |
| Input Offset Current | $I_{IO}$ | $V_{CM}$ = 0V | | - | 2.0 | 30 | - | 3.0 | 50 | - | 3.0 | 50 | nA |
| Input Bias Current | $I_{BIAS}$ | $V_{CM}$ = 0V | | - | 40 | 150 | - | 40 | 250 | - | 40 | 250 | nA |
| Input Common-Mode Voltage Range | $V_{I(R)}$ | Note1 | | 0 | - | $V_{CC}$ -1.5 | 0 | $V_{CC}$ -1.5 | - | 0 | - | $V_{CC}$ -1.5 | V |
| Supply Current | $I_{CC}$ | $R_L$ = ∞, $V_{CC}$ = 30V (LM2902, $V_{CC}$=26V) | | - | 1.0 | 3 | - | 1.0 | 3 | - | 1.0 | 3 | mA |
| | | $R_L$ = ∞, $V_{CC}$ = 5V | | - | 0.7 | 1.2 | - | 0.7 | 1.2 | - | 0.7 | 1.2 | mA |
| Large Signal Voltage Gain | $G_V$ | $V_{CC}$ = 15V, $R_L$=2kΩ, $V_{O(P)}$ = 1V to 11V | | 50 | 100 | - | 25 | 100 | - | 25 | 100 | - | V/mV |
| Output Voltage Swing | $V_{O(H)}$ | Note1 | $R_L$ = 2kΩ | 26 | - | - | 26 | - | - | 22 | - | - | V |
| | | | $R_L$=10kΩ | 27 | 28 | - | 27 | 28 | - | 23 | 24 | - | V |
| | $V_{O(L)}$ | $V_{CC}$ = 5V, $R_L$=10kΩ | | - | 5 | 20 | - | 5 | 20 | - | 5 | 100 | mV |
| Common-Mode Rejection Ratio | CMRR | - | | 70 | 85 | - | 65 | 75 | - | 50 | 75 | - | dB |
| Power Supply Rejection Ratio | PSRR | - | | 65 | 100 | - | 65 | 100 | - | 50 | 100 | - | dB |
| Channel Separation | CS | f = 1kHz to 20kHz (Note2) | | - | 120 | - | - | 120 | - | - | 120 | - | dB |
| Short Circuit to GND | $I_{SC}$ | $V_{CC}$ = 15V | | - | 40 | 60 | - | 40 | 60 | - | 40 | 60 | mA |
| Output Current | $I_{SOURCE}$ | $V_{I(+)}$ = 1V, $V_{I(-)}$ = 0V, $V_{CC}$ = 15V, $V_{O(P)}$ = 2V | | 20 | 40 | - | 20 | 40 | - | 20 | 40 | - | mA |
| | $I_{SINK}$ | $V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V, $V_{CC}$ = 15V, $V_{O(P)}$ = 2V | | 10 | 13 | - | 10 | 13 | - | 10 | 13 | - | mA |
| | | $V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V, $V_{CC}$ = 5V, $V_{O(R)}$ = 200mV | | 12 | 45 | - | 12 | 45 | - | - | - | - | μA |
| Differential Input Voltage | $V_{I(DIFF)}$ | - | | - | - | $V_{CC}$ | - | - | $V_{CC}$ | - | - | $V_{CC}$ | V |

**Note :**

1. $V_{CC}$=30V for LM224 and LM324 , $V_{CC}$ = 26V for LM2902
2. This parameter, although guaranteed, is not 100% tested in production.

## Electrical Characteristics (Continued)

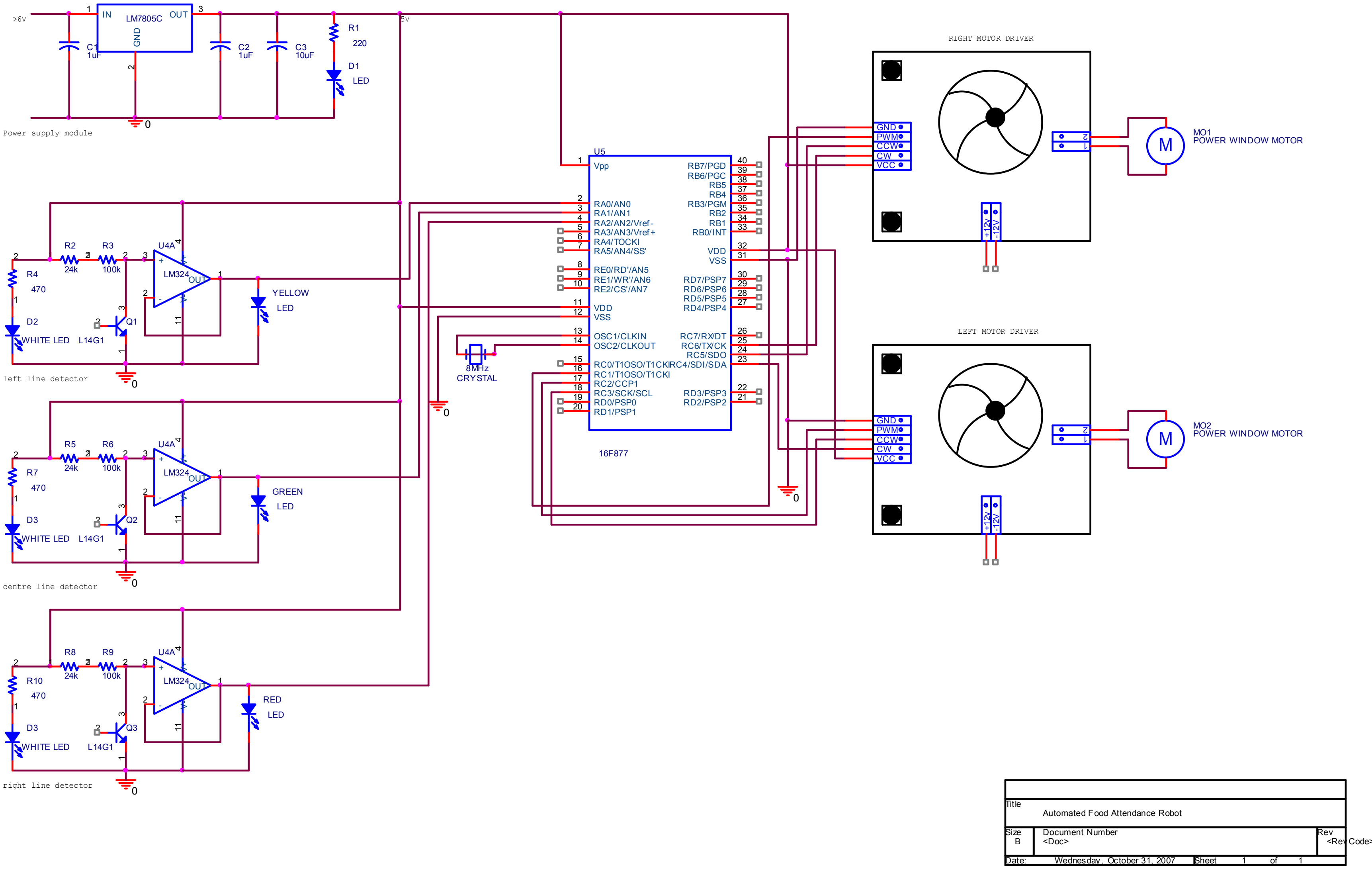
($V_{CC}$ = 5.0V, $V_{EE}$ = GND, unless otherwise specified)
The following specification apply over the range of -25°C ≤ $T_A$ ≤ +85°C for the LM224A; and the 0°C ≤ $T_A$ ≤ +70°C for the LM324A

| Parameter | Symbol | Conditions | | LM224A | | | LM324A | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| Input Offset Voltage | $V_{IO}$ | $V_{CM}$ = 0V to $V_{CC}$ -1.5V $V_{O(P)}$ = 1.4V, $R_S$ = 0Ω (Note1) | | - | - | 4.0 | - | - | 5.0 | mV |
| Input Offset Voltage Drift | $\Delta V_{IO}/\Delta T$ | $R_S$ = 0Ω (Note2) | | - | 7.0 | 20 | - | 7.0 | 30 | μV/°C |
| Input Offset Current | $I_{IO}$ | $V_{CM}$ = 0V | | - | - | 30 | - | - | 75 | nA |
| Input Offset Current Drift | $\Delta I_{IO}/\Delta T$ | $R_S$ = 0Ω (Note2) | | - | 10 | 200 | - | 10 | 300 | pA/°C |
| Input Bias Current | $I_{BIAS}$ | - | | - | 40 | 100 | - | 40 | 200 | nA |
| Input Common-Mode Voltage Range | $V_{I(R)}$ | Note1 | | 0 | - | $V_{CC}$ -2.0 | 0 | - | $V_{CC}$ -2.0 | V |
| Large Signal Voltage Gain | $G_V$ | $V_{CC}$ = 15V, $R_L$= 2.0kΩ | | 25 | - | - | 15 | - | - | V/mV |
| Output Voltage Swing | $V_{O(H)}$ | Note1 | $R_L$ = 2kΩ | 26 | - | - | 26 | - | - | V |
| | | | $R_L$ = 10kΩ | 27 | 28 | - | 27 | 28 | - | V |
| | $V_{O(L)}$ | $V_{CC}$ = 5V, $R_L$= 10kΩ | | - | 5 | 20 | - | 5 | 20 | mV |
| Output Current | $I_{SOURCE}$ | $V_{I(+)}$ = 1V, $V_{I(-)}$ = 0V $V_{CC}$ = 15V, $V_{O(P)}$ = 2V | | 10 | 20 | - | 10 | 20 | - | mA |
| | $I_{SINK}$ | $V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V $V_{CC}$ = 15V, $V_{O(P)}$ = 2V | | 5 | 8 | - | 5 | 8 | - | mA |
| Differential Input Voltage | $V_{I(DIFF)}$ | - | | - | - | $V_{CC}$ | - | - | $V_{CC}$ | V |

**Note:**
1. $V_{CC}$=30V for LM224A and LM324A.
2. These parameters, although guaranteed, are not 100% tested in production.

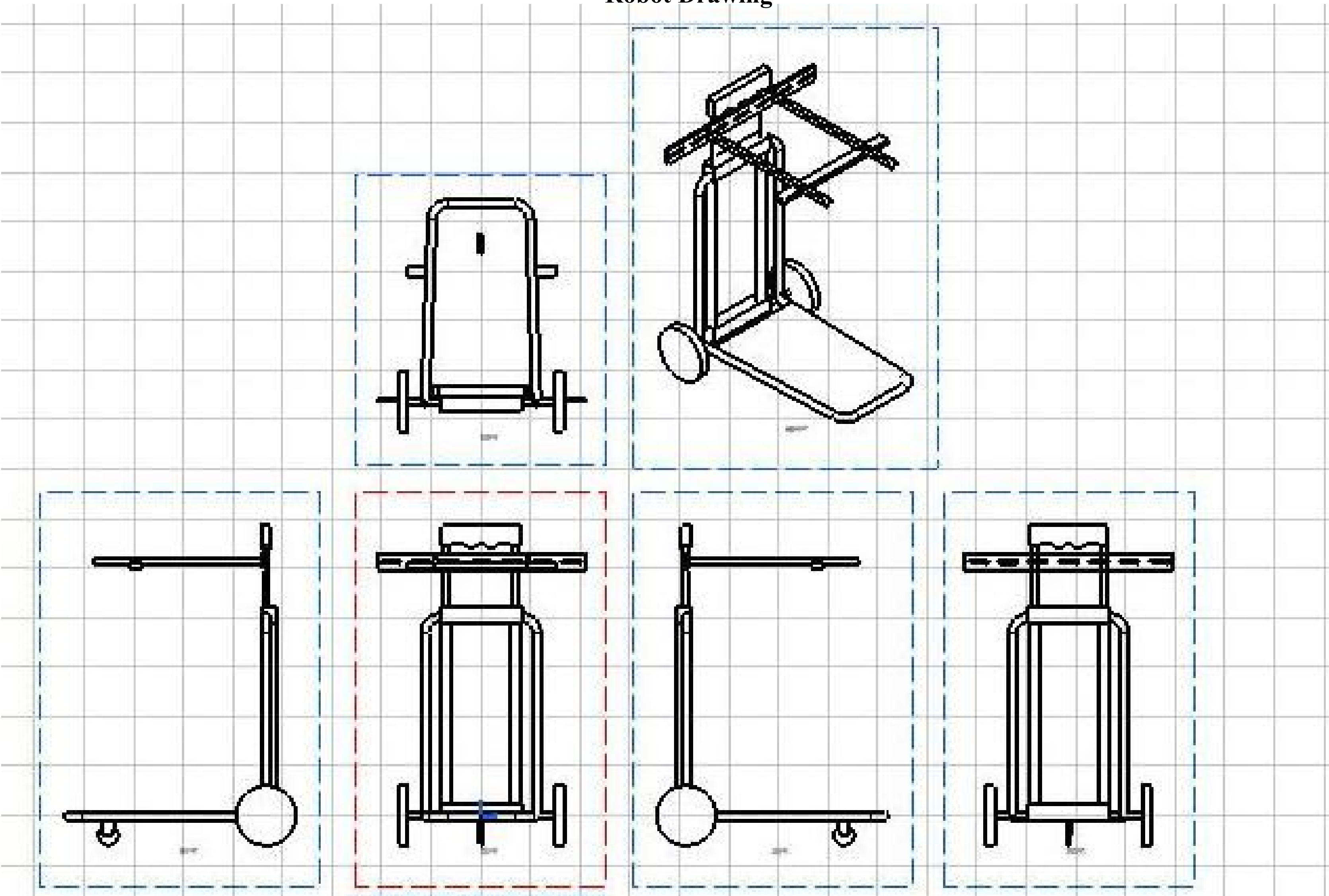
# Appendix E

## Full Circuit Diagram

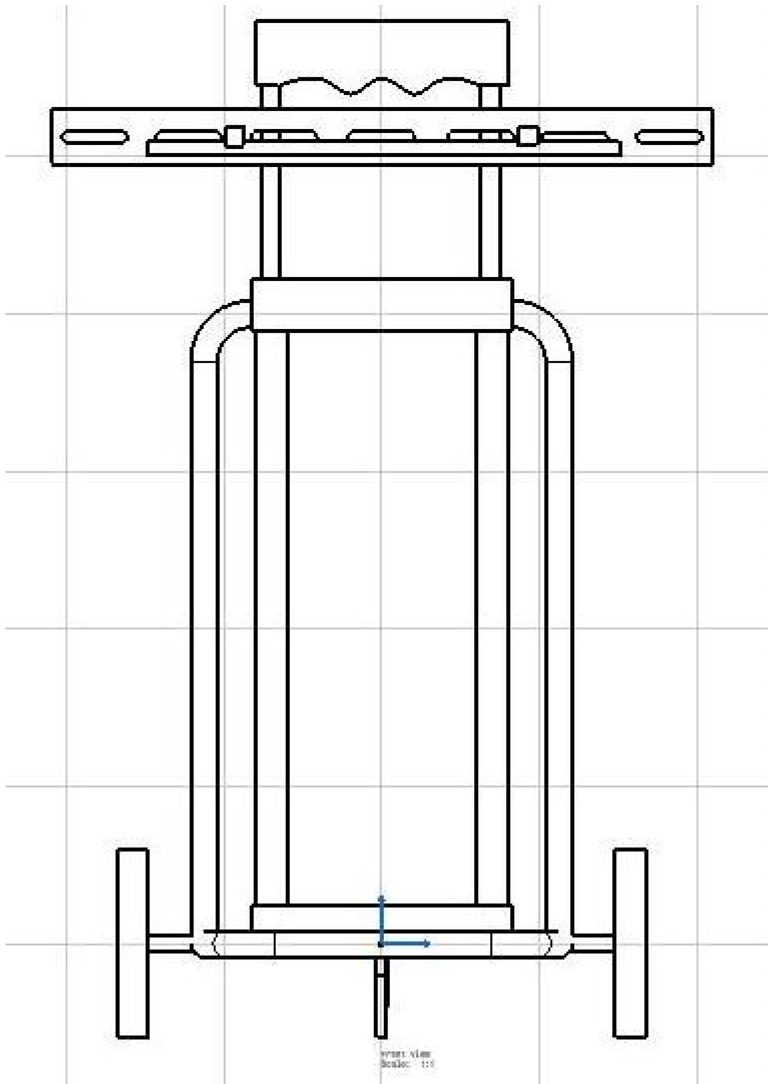


**Appendix F**

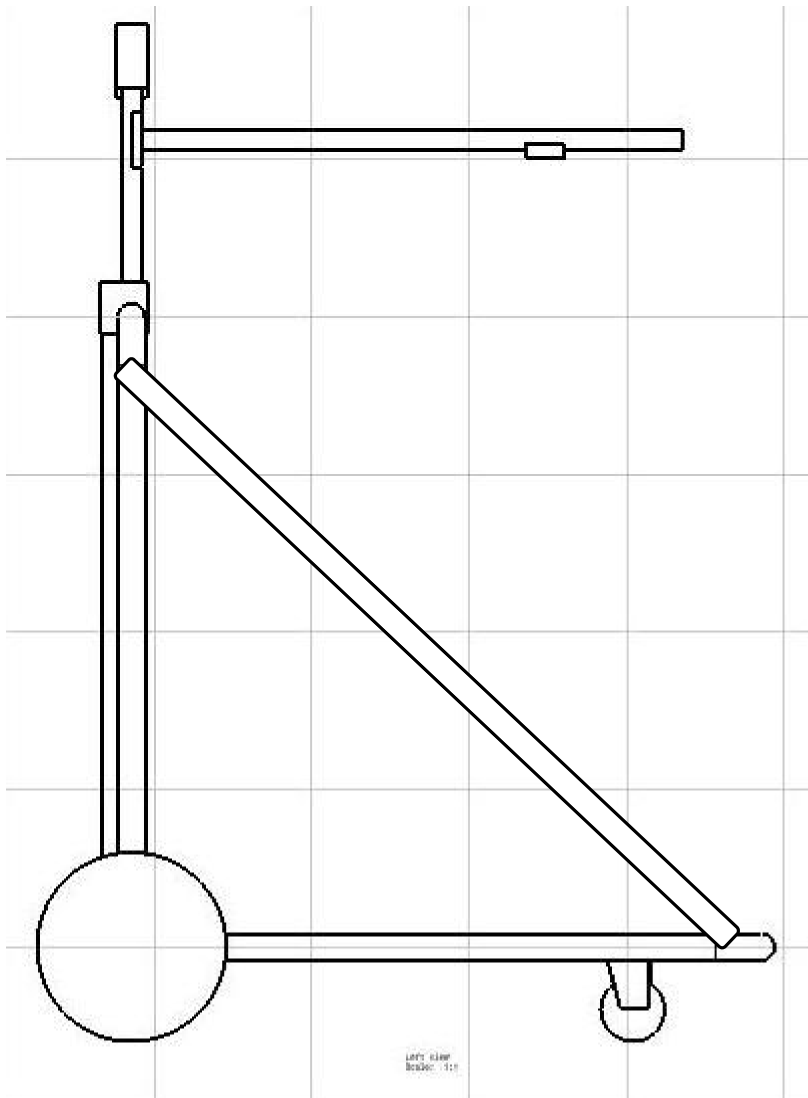
**Robot Drawing**



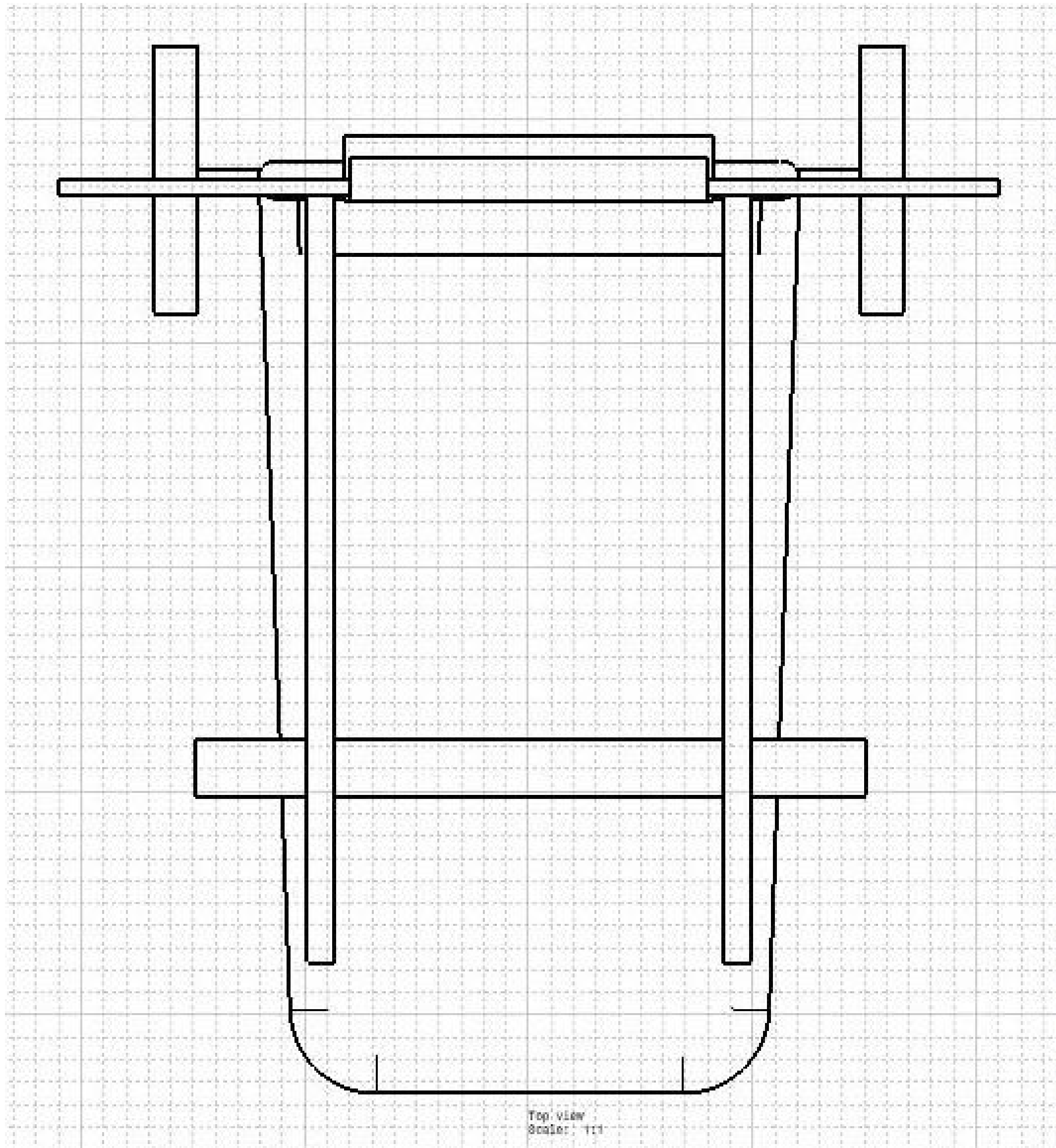
All View



Front View



Side View




Top View

## 11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the other devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, VSS, RA2, or RA3.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the voltage reference), or as digital I/O.

Additional information on using the A/D module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

**REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | — | ADON |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0**: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)(1)
110 = channel 6, (RE1/AN6)(1)
111 = channel 7, (RE2/AN7)(1)

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
If ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1 **Unimplemented**: Read as '0'

bit 0 **ADON**: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

**Note 1:** These channels are not available on PIC16F873/876 devices.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)**

| U-0 | U-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0**: A/D Port Configuration Control bits:

| PCFG3: PCFG0 | AN7[(1)] RE2 | AN6[(1)] RE1 | AN5[(1)] RE0 | AN4 RA5 | AN3 RA3 | AN2 RA2 | AN1 RA1 | AN0 RA0 | VREF+ | VREF- | CHAN/ Refs[(2)] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000 | A | A | A | A | A | A | A | A | VDD | VSS | 8/0 |
| 0001 | A | A | A | A | VREF+ | A | A | A | RA3 | VSS | 7/1 |
| 0010 | D | D | D | A | A | A | A | A | VDD | VSS | 5/0 |
| 0011 | D | D | D | A | VREF+ | A | A | A | RA3 | VSS | 4/1 |
| 0100 | D | D | D | D | A | D | A | A | VDD | VSS | 3/0 |
| 0101 | D | D | D | D | VREF+ | D | A | A | RA3 | VSS | 2/1 |
| 011x | D | D | D | D | D | D | D | D | VDD | VSS | 0/0 |
| 1000 | A | A | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 6/2 |
| 1001 | D | D | A | A | A | A | A | A | VDD | VSS | 6/0 |
| 1010 | D | D | A | A | VREF+ | A | A | A | RA3 | VSS | 5/1 |
| 1011 | D | D | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 4/2 |
| 1100 | D | D | D | A | VREF+ | VREF- | A | A | RA3 | RA2 | 3/2 |
| 1101 | D | D | D | D | VREF+ | VREF- | A | A | RA3 | RA2 | 2/2 |
| 1110 | D | D | D | D | D | D | D | A | VDD | VSS | 1/0 |
| 1111 | D | D | D | D | VREF+ | VREF- | D | A | RA3 | RA2 | 1/2 |

A = Analog input D = Digital I/O

**Note 1:** These channels are not available on PIC16F873/876 devices.
**2:** This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared x = Bit is unknown |

The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D result register pair, the GO/DONE bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 11-1.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

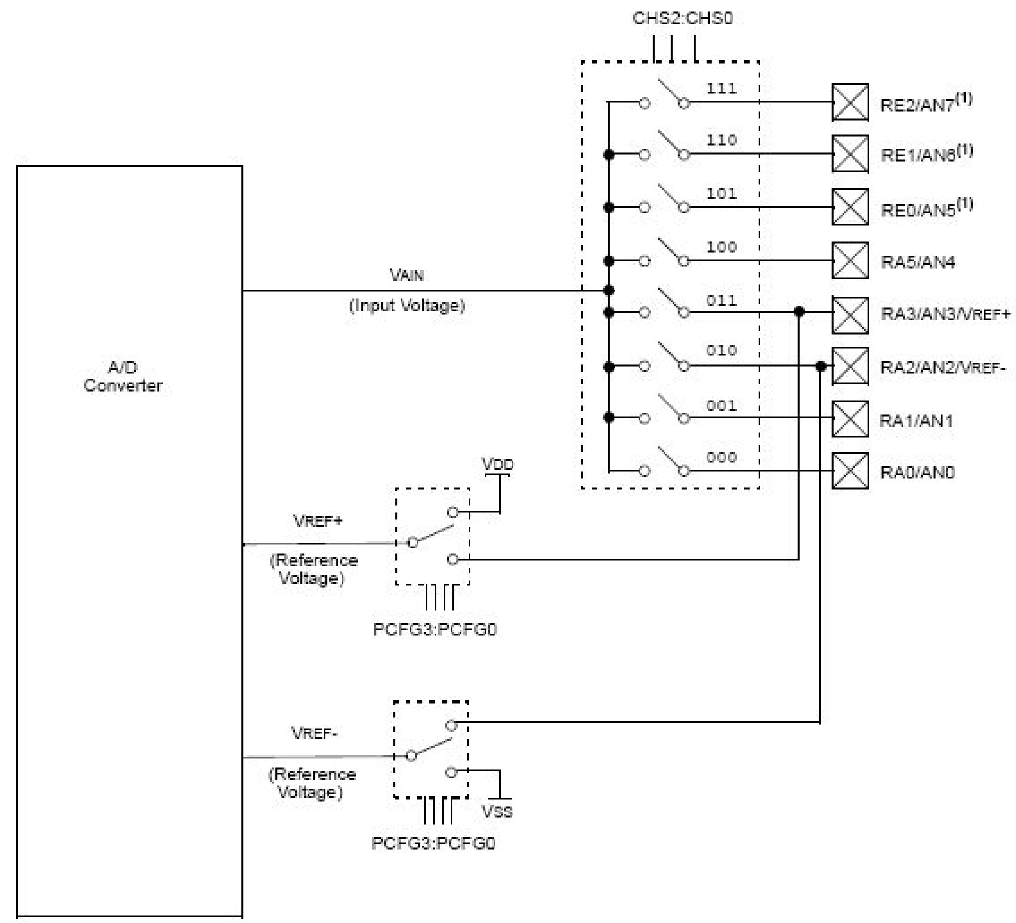
To determine sample time, see Section 11.1. After this acquisition time has elapsed, the A/D conversion can be started.

 © 2001 Microchip Technology Inc.

These steps should be followed for doing an A/D Conversion:

1. Configure the A/D module:
   - Configure analog pins/voltage reference and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set PEIE bit
   - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/DONE bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/DONE bit to be cleared (with interrupts enabled); OR
   - Waiting for the A/D interrupt
6. Read A/D result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For the next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2TAD is required before the next acquisition starts.

**FIGURE 11-1: A/D BLOCK DIAGRAM**




Note 1: Not available on PIC16F873/876 devices.

© 2001 Microchip Technology Inc.

## 11.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 11-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 11-2. **The maximum recommended impedance for analog sources is 10 kΩ.** As the impedance is decreased, the acquisition time may be decreased.

After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 11-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the PICmicro™ Mid-Range Reference Manual (DS33023).

**EQUATION 11-1: ACQUISITION TIME**

$$
\begin{aligned}
T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\
&= T_{AMP} + T_C + T_{COFF} \\
&= 2\mu s + T_C + [(\text{Temperature} - 25^\circ C)(0.05\mu s/^\circ C)] \\
T_C &= C_{HOLD}\,(R_{IC} + R_{SS} + R_S)\,\ln(1/2047) \\
&= -120pF\,(1k\Omega + 7k\Omega + 10k\Omega)\,\ln(0.0004885) \\
&= 16.47\mu s \\
T_{ACQ} &= 2\mu s + 16.47\mu s + [(50^\circ C - 25^\circ C)(0.05\mu s/^\circ C)] \\
&= 19.72\mu s
\end{aligned}
$$

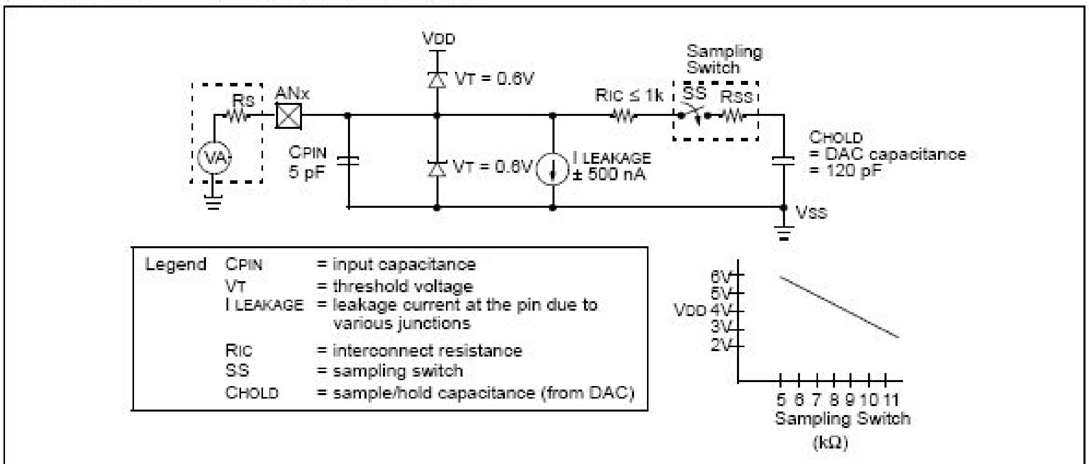
**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

**4:** After a conversion has completed, a 2.0TAD delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

**FIGURE 11-2: ANALOG INPUT MODEL**





© 2001 Microchip Technology Inc.

## 11.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2TOSC
- 8TOSC
- 32TOSC
- Internal A/D module RC oscillator (2-6 μs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 μs.

Table 11-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

**TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))**

| AD Clock Source (TAD) | | Maximum Device Frequency |
|---|---|---|
| **Operation** | **ADCS1:ADCS0** | **Max.** |
| 2TOSC | 00 | 1.25 MHz |
| 8TOSC | 01 | 5 MHz |
| 32TOSC | 10 | 20 MHz |
| RC(1, 2, 3) | 11 | **(Note 1)** |

**Note 1:** The RC source has a typical TAD time of 4 μs, but can vary between 2-6 μs.
**2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for SLEEP operation.
**3:** For extended voltage devices (LC), please refer to the Electrical Characteristics (Sections 15.1 and 15.2).

## 11.3 Configuring Analog Port Pins

The ADCON1 and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

**Note 1:** When reading the port register, any pin configured as an analog input channel will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

**2:** Analog levels on any pin that is defined as a digital input (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the device specifications.

© 2001 Microchip Technology Inc.

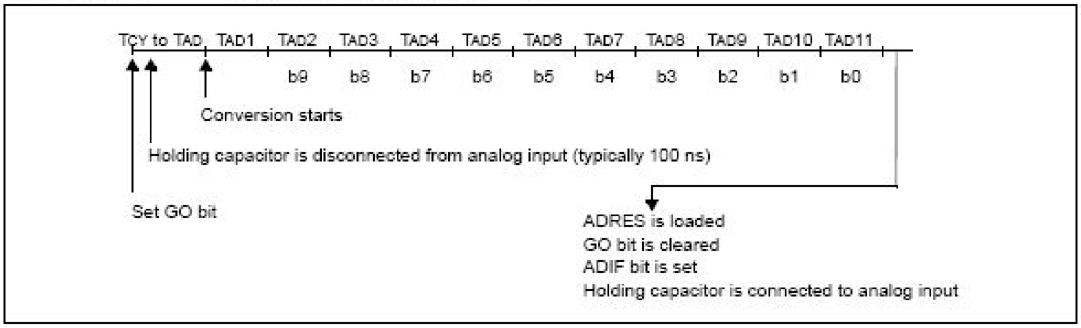
## 11.4 A/D Conversions

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, acquisition on the selected channel is automatically started. The GO/$\overline{\text{DONE}}$ bit can then be set to start the conversion.

In Figure 11-3, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

| | |
|---|---|
| **Note:** | The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D. |

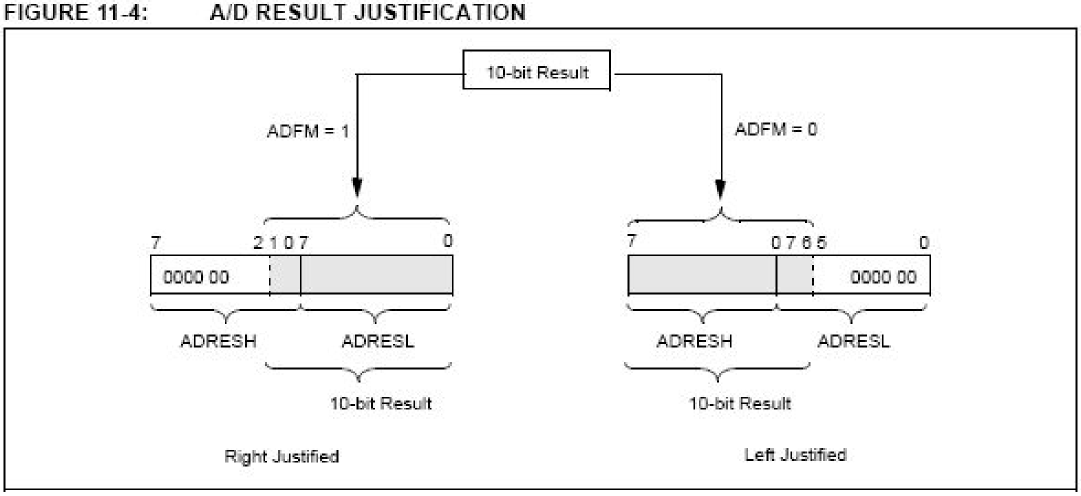
**FIGURE 11-3: A/D CONVERSION TAD CYCLES**




### 11.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16-bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 11-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's'. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

**FIGURE 11-4: A/D RESULT JUSTIFICATION**




 © 2001 Microchip Technology Inc.

**APPENDIX H**
**Full Program Using PIC**

```
'*************************************************************
'*  Name    : Automated Food Attendance Robot Program         *
'*  Author  : Nur Izzati Bt Abdullah                          *
'*  Notice  : Copyright (c) 2007 [select VIEW...EDITOR OPTIONS]  *
'*          : All Rights Reserved                             *
'*  Date    : 10/22/2007                                      *
'*  Version : 1.0                                             *
'*  Notes   :                                                 *
'*          :                                                 *
'*************************************************************
DEFINe OSC 8
' sight sensors conf
DEFINE ADC_BITS 8
define ADC_SAMPLEUS 10
define ccp1_reg portc
define ccp1_bit 2
define ccp2_reg portc
define ccp2_bit 1
ADC VAR BYTE
SIGHT VAR BYTE
sense_limit var byte
motor_flag var byte
R var byte
C var byte
L var byte
TRISA=255
TRISC=0
ADCON1=1
sense_limit = 80 'asal 90'
'***************main program start*****************
start:
gosub sensor_scan
gosub sight_motor
goto start
sensor_scan:
    ADCin 0,adc 'asal porta.1 change to porta.0 and now the letest'
    IF ADC>=SENSE_LIMIT THEN 'asal >='
      R=1
      'gosub turn_right'    'connectkan led kat portb.0 sbg indicator'
    ELSE
      R=0
      'gosub motor_stop'
    Endif
      'asal ade pause 10'
    adcin 1,adc
    if adc>=sense_limit then
      C=1
```

```
        else
           C=0
        endif
        ADCIN 2,ADC
        IF ADC>=sense_limit then
           L=1
        else
           L=0
        endif
return
'right motor=ccp2=1, eyes from free drive tyre'
'left motor=ccp1=0,  eyes from free drive tyre'

turn_right:
    hpwm 1,0,5000
    hpwm 0,255,5000
    high portc.6
    low  portc.5
    high portc.4
    low  portc.3
return

turn_left:
    hpwm 1,255,5000
    hpwm 0,0,5000
    high portc.6
    low  portc.5
    high portc.4
    low  portc.3
return

sight_motor:
    if C=1 then
        gosub motor_run
        return
    endif
    IF R=1 then
        gosub turn_right
        return
    endif
         if L=1 then
           gosub turn_left
           return
         endif
    if R=0 and c=0 and L=0 then
        gosub motor_stop
        return
```

```
  endif
return

motor_run:
  hpwm 1,255,5000
  hpwm 0,255,5000
  high portc.6
  low  portc.5
  high portc.4
  low  portc.3
return

motor_stop:
  hpwm 1,0,5000
  hpwm 0,0,5000
  high portc.6
  low  portc.5
  high portc.4
  low  portc.3
return
```