OBSTACLE AVOIDANCE LEGGED ROBOT.

MOHAMED FAIZAN BIN BASHEER AHMAD

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

Faculty of Electrical & Electronic Engineering
University Malaysia Pahang

NOVEMBER 2008

DECLARATION

"All the trademark and copyrights use here in 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 : MOHAMED FAIZAN BIN BASHEER AHMAD.

Date : 21 NOVEMBER 2008.

DEDICATION

Specially dedicate to

My beloved parents, brothers and sisters.

ACKNOWLEDGEMENT

First, I would like to express highest gratitude to my supervisor, Encik Saifudin b. Razali for the guidance and co-operation that been given throughout the progress and completion of this project.

I also deeply thank to my family who's always giving a motivational support and show me a guidance to pursue my studies at University Malaysia Pahang. Thanks for their encouragement, support, love, my brothers, sister and many more. ALHAMDULILLAH.

Finally, my great appreciation to my dear housemates that given me so many opinion and encouragement till I don't know which one to comprehend, especially Hanis , Khawarizmi and Mohd Noor thanks for their brilliant idea and my class mate whom involve directly or indirectly with this project. Last but not least the dedication of a thousand of thank to dear Hasinah that is a special one to me that had always been a backbone for me and gave plenty of motivation, moral support and so on. Finally thank you Allah since through your blessings I managed to do all in time. Thank You very Much.

ABSTRACT

Many mobile robots require an operator's vision and intelligence for guidance and navigation. Animals use sensory systems such as hearing, and tactile to move freely through their environment. The aim of this project is to develop an avoidance behaviors program for a mobile robot that consists of 8 servo motors that been employed at 4 legs. Each leg contains 2 servo motors, one for X-axis and one for Y-axis. A PIC Microcontroller has been implemented to act as a brain for the robot that controls the walking and turning algorithm. Ultrasonic sensor was also developed to act as an 'eye' to the system and tells the brain about existence of obstacle in front. As a resultant, the obstacle avoidance legged robot system is been successfully developed that allows and will navigate the robot to move through the environment freely. But there is a certain limitation for the robot such as the wideness of the area, type of obstacle and surface.

ABSTRAK

Di dalam dunia permodenan dan kontemporari seperti ini robot memerlukan operator dan kebijaksanaan wawasan untuk tujuan panduan hala tujunya. Demikian adalah cara yang digunakan haiwan iaitu sistem-sistem deria pendengaran, dan sentuhan untuk bergerak dengan lebih bebas menerusi alam sekitar mereka. Matlamat projek ini adalah untuk membangunkan satu pergerakan pengelakan atur cara untuk satu kepergerakan robot yang mengandungi 8 servo motor yang berfungsi di setiap kaki. Keseluruhan robot itu mempunyai 4 kaki dan pada tiap kaki mengandungi 2 servo motor , satu untuk pergerakkan paksi X dan satu lagi untuk pergerakkan paksi Y. Satu PIC Microcontroller telah digunakan dan ia berfungsi sebagai satu otak untuk mengawal robot itu dengan wujudnya algoritma berjalan dan algoritma selekoh. Deria ultrasonik juga dibuat supaya ia berfungsi sebagai satu 'mata' bagi robot itu.

TABLE OF CONTENT

LIST OF CHAPTER

LIST OF TABLES

LIST OF FIGURES.

LIST OF SYMBOL.

ms	-	Millisecond.
I/O	-	Input Output.
IC	-	Integrated Circuit.
R	-	Resistor.
C	-	Capacitor.
Cθ	-	cos (θ)
Sθ	-	sin (θ)
LED	-	Light Emitter Diode.
IR	-	Infrared.
k	-	Kilogram.
V	-	Voltage.
mA	-	mili ampere.
kg	-	kilo gram.
PWM	-	Pulse Width Modulation
ADC	-	Analog Digital Converter
PIC	-	Programmable Interface Controller

LIST OF APPENDICES.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW.

Robots are designed to be controlled by a controller, computer or similar devices. Basically mobile robots require an operator's vision and intelligence for guidance and navigation. The movement for the robot itself has many methods such as wheels, legs and many more. This movement method is to ensure the smoothness of moving in different type of surfaces.

A simple wheeled vehicle is easy in mechanical design, controlling, and the construction part. But it doesn't work efficiently in all kind of surface. On a rough terrain, it performs poorly. The radius of a wheel could pass only a certain height of obstacle. To pass most of the obstacle that it meets, larger wheel radius need to be designed. However, this approach is impractical in many cases.

On the other side, legged robots are more capable of moving across rough terrain. That's why the legged locomotion became a research of interest. But the legged robots are much more challenging to control compared wheel robots. Each leg consists of at least two motors. Controlling the one motor for left and right and another motor for up and down is difficult. An algorithm for walking must be developed in order the robot to walk. As a result, legged robots movement must be carefully studied and be controlled in such way so that it could stand and walk in a stable fashion.

1.2 OBJECTIVE OF THE PROJECT.

The aim of this project is to develop an avoidance behaviors program for a mobile robot that consists of 4 legs that employs 8 servo motors. A PIC Microcontroller is been implemented to act as the brain for the robot that controls the walking and turning algorithm. The system and the programming will be able to control the movement of the legged so the robot able to walk straight ahead, make left turns and avoid obstacle.

1.3 SCOPE OF PROJECT.

Several scopes have been outlined to achieve this project. Scope of this project includes developing a program using high level language PIC Basic Pro for the PIC16F877A microcontroller, constructing the hardware module for it, attachment of servo motors and integration of ultrasonic sensor with the microcontroller. Developing an algorithm program for the robot to walk ahead, make a left turn and is capable to avoid obstacle.

1.4 OUTLINE OF THESIS.

This thesis consists of five chapters. This chapter is discussed about the background of robot, the objective, and scope of the project. In chapter 2 is more towards reviewing about literature study. It discusses about the function of the servo motor and it could be operated using PWM, about details of PIC16F877A microcontroller, about other robotic research that been done and about various type of sensor. In chapter 3 is been discussed about methodology for hardware and software implementation into the project. The result, analysis and discussion will be on the chapter 4. Finally, chapter 5 discusses the conclusion of the project and future work that can be done to improvise this project.

CHAPTER 2

THEORY AND LITERATURE REVIEW

2.1 INTRODUCTION.

This chapter is upon the study on servo motor, PIC microcontroller, robots analysis and sensors. Servo motor was reviewed upon controlling it using PWM and how it could be used as a device that controls a degree of freedom in a quadruped robot. The PIC microcontroller discussed more towards the capability of it and thorough detail on the functions and the operations of it. Type of sensors compared in this chapter.

2.2 SERVO MOTORS.

Servo motors are commonly used in robotics especially that uses leg for movement purpose for the robot. This was because they are light-weighted, compact and durable. Since their control and power electronics were all built-in, interfacing hardware requirement is much simplified. Compared to toy motors, the torque of servo motors are high so the servo motors better suited to robot development.

Each servo motor has three wires, that is yellow, red and black. As shown in Figure 2.0 red is for power, black is for ground, and yellow is for pulse-width modulation (PWM) pulsing. This pulsing controls the position which the shaft of a servo motor should rotate.

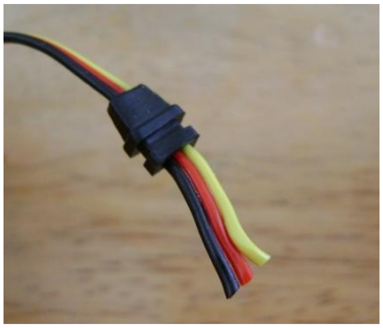


Figure 2.0: Wires for the servo motor, black (ground), red (power), yellow (PWM)

Figure 2.1 shows the servo motor dimension that been used in this project, the servo motor that been used is Hitec HS -322HD servo motor. The specification of the servo is show in Table 2.0.

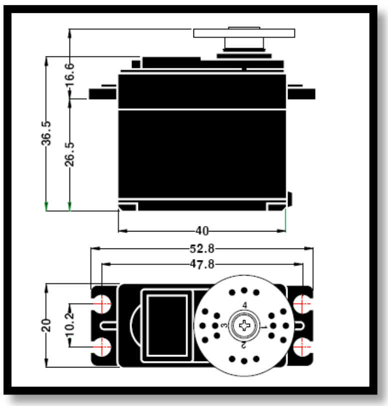


Figure 2.1: Servo motor Hitec HS -322HD dimension. (The dimension in mm)

Table 2.0: Hitec HS-322HD servo motor specification [1].

Operating voltage range	4.8V to 6V	
Test Voltage	**4.8V**	**6V**
Operating speed	0.19s/ 60° at no load	0.15s/ 60° at no load
Stall Torque	3kg. cm	3.7kg. cm
Ideal current at stopped	7.4mA	7.7mA
Running current	160mA/ 60° at no load	180mA/ 60° at no load
Stall current	700mA	800mA
Motor type	Cored metal brush	
Potentiometer type	4 slide/ direct drive	
Amplifier type	Analog controller and transistor drive.	
Dimension	40 x 20 x 36.5mm	
Weight	43g	
Ball bearing	Top/ Resin bushing	
Gear material	2 heavy duty resin	
Horn gear spline	24 segments	
Connector wire length	300mm	

2.2.1 PULSE WIDTH MODULATION FOR SERVO

Pulse-width modulated (PWM) signal is a train of pulses with varying width. Each servo motor does have its own unique pulsing width. But these listed configurations are for the servos that have been used in this project that is HS-322HD. The full duty cycle or called period for a whole pulsing must be 20ms. Figure 2.2, shows the movement of the degree upon the timeline (PWM) of the duty cycle produced. The time of 0.6ms, 1.5ms and 2.4 ms signaled the motor shaft to turn to 0 degree, 90 degrees and 180 degrees respectively. The motor shaft could not go any further than below 0 degree and above 180 degree range unless modifications were made to the servo motor.

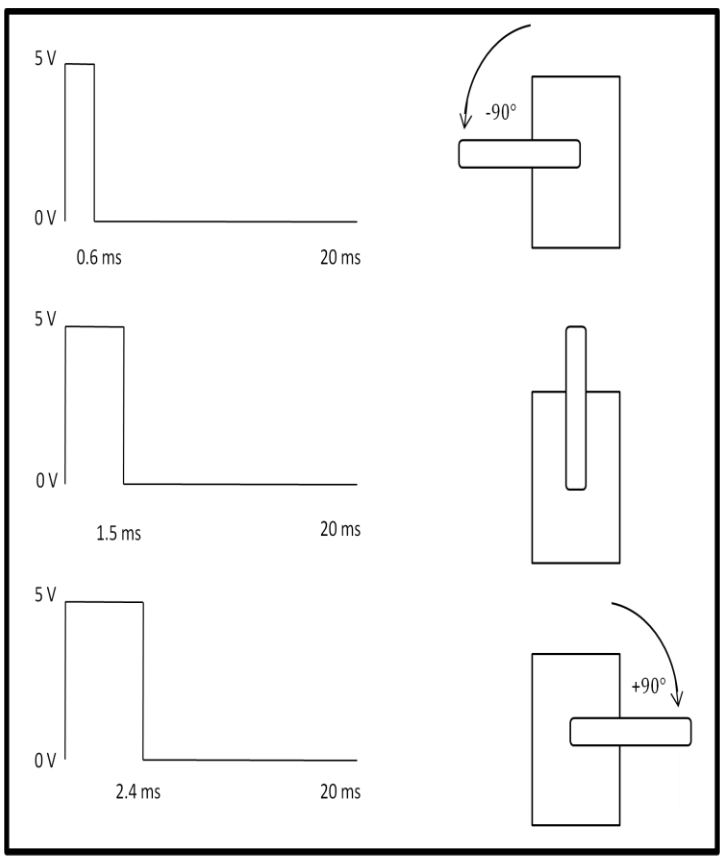


Figure 2.2: Position of motor shaft when PWM is generated.

2.2.2 INSIDE SERVO

The servo motor has some control circuits and a potentiometer (a variable resistor, a.k.a. pot) that is connected to the output shaft. In the Figure 2.3, the pot can be seen on the right side of the circuit board. This pot allows the control circuitry to monitor the current angle of the servo motor. If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor the correct direction until the angle is correct. A normal servo is used to control an angular motion of between 0 and 180 degrees. A normal servo is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear.

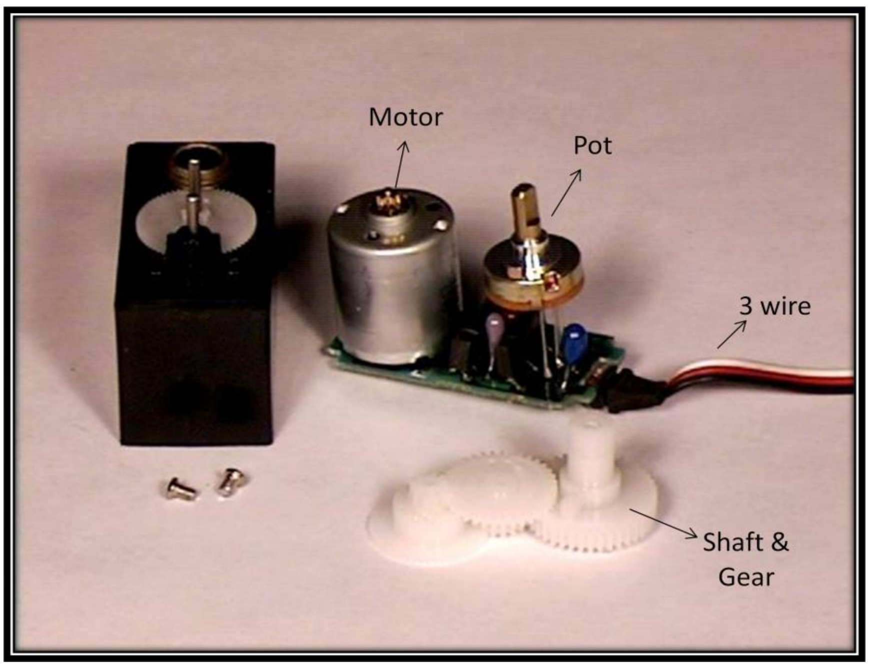


Figure 2.3: The inner part of servos.

2.3 PIC MICROCONTROLLER

For this project, the controller is used as a brain for the whole system. From the input from sensor until the output for the servo motor is controlled totally using this PIC microcontroller. Describing about the PIC microcontroller and in the mean time will be given more understanding upon employing this controller. Almost all type of PIC microcontroller is included in a class of 8-bit microcontrollers of RISC architecture. Basically, the PIC architecture is minimized to be a simpler item but it still operates at the same function. The Harvard architecture is a newer concept than von-Neumann. It was designed as a response for the need to speed up the work of a microcontroller. In Harvard architecture, data bus and address bus are separate. Thus, the data will flow directly through the central processing unit and the address bus is neglected. This greater flow of data will impact for a greater speed of work. Besides that, the architecture will involve for a small number of a fixed length instruction. It means the instruction is not to have to be 8-bit words but it can uses 14 bits for instructions which allows for all instruction to be one word instructions. Microcontrollers with Harvard architecture are called "RISC microcontrollers". RISC is a short form for Reduced Instruction Set Computer. Microcontrollers with von-Neumann's architecture are called 'CISC microcontrollers'. CISC is a short form for Complex Instruction Set Computer. Same as discussion before, RISC microcontroller has a reduced set of instructions, maybe 35 instructions for one cycle. If we compared it with Intel’s and Motorola’s microcontroller, it has over hundred instructions. As a simplified point, we can say that the features of PIC microcontroller are:

(i) Separate code and data spaces (Harvard architecture).
(ii) A small number of fixed length instructions.
(iii) Most instructions are single cycle execution (4 clock cycles), with single delay cycles upon branches and skips.
(iv) All RAM locations function as registers as both source and/or destination of math and other functions.
(v) A hardware stack for storing return addresses.

(vi) A fairly small amount of addressable data space (typically 256 bytes), extended through banking.

(vii) Data space mapped CPU, port and peripheral registers.

(viii) The program counter is also mapped into the data space and writable.

The result for PIC microcontroller reaches of 2:1 in code compression and 4:1 in speed in relation to other 8-bit microcontrollers in its class. Generally, PIC microprocessor divides to 6 parts. Those are program memory, EEPROM, RAM, PORTA and PORTB, free-run timer and central processing unit. For more detailed explained, we specified the function of each part below.

(i) **Program memory** (FLASH) - for storing a written program. Using flash technology, the memory can be programmed and cleared more than once. It makes this microcontroller suitable for device development.

(ii) **EEPROM -** data memory that needs to be saved when there is no supply. It is usually used for storing important data that must not be lost if power supply suddenly stops. For instance, one such data is an assigned temperature in temperature regulators. If during a loss of power supply this data was lost, we would have to make the adjustment once again upon return of supply. Thus our device looses on self-reliance.

(iii) **RAM** - data memory used by a program during its execution. In RAM, it are stored all inter-results or temporary data during run-time.

(iv) **PORTA and PORTB** are physical connections between the microcontroller and the outside world. Port A has five, and port B has eight pins.

(v) **FREE-RUN TIMER** is an 8-bit register inside a microcontroller that works independently of the program. On every fourth clock of the oscillator it increments its value until it reaches the maximum (255), and then it starts counting over again from zero. As we know the exact timing between each two increments of the timer contents, timer can be used for measuring time which is very useful with some devices.

(vi) **CENTRAL PROCESSING UNIT** has a role of connective element between other blocks in the microcontroller. It coordinates the work of other blocks and executes the user program.

Same as other microcontroller, the main starter for PIC microcontroller is the clock or instruction cycle. It is attached to the PIC microcontroller as an external component called as “oscillator”. We connect the oscillator with the pin OSC1 and clock will enter the microcontroller where the clock will be divided by the internal circuit of a microcontroller into four even clock Q1, Q2, Q3, Q4 which do not overlap. These four clocks make up one instruction cycle (also called machine cycle) during which one instruction is executed. Execution of instruction starts by calling an instruction that is next in string. Instruction is called from program memory on every Q1 and is written in instruction register on Q4. Decoding and execution of instruction are done between the next Q1 and Q4 cycles. On the following diagram we can see the relationship between instruction cycle and clock of the oscillator (OSC1) as well as that of internal clocks Q1-Q4. Program counter (PC) holds information about the address of the next instruction.

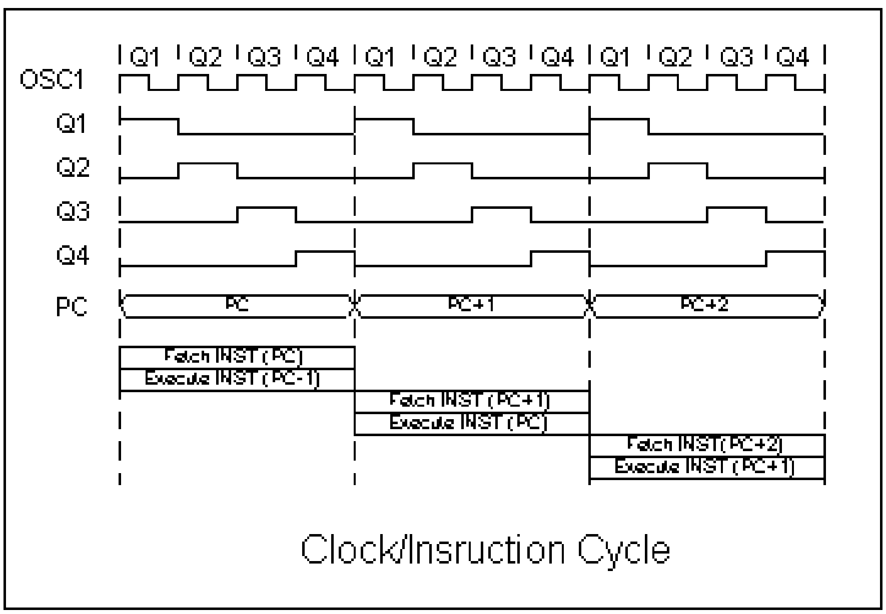


Figure 2.4: Clock/instruction Cycle

Furthermore, PIC16877 will be applied as the specified microcontroller. In general, the characteristic of PIC16877 are same as explained before but the difference is amount of number pin. PIC16877 has a total of 40 pins.

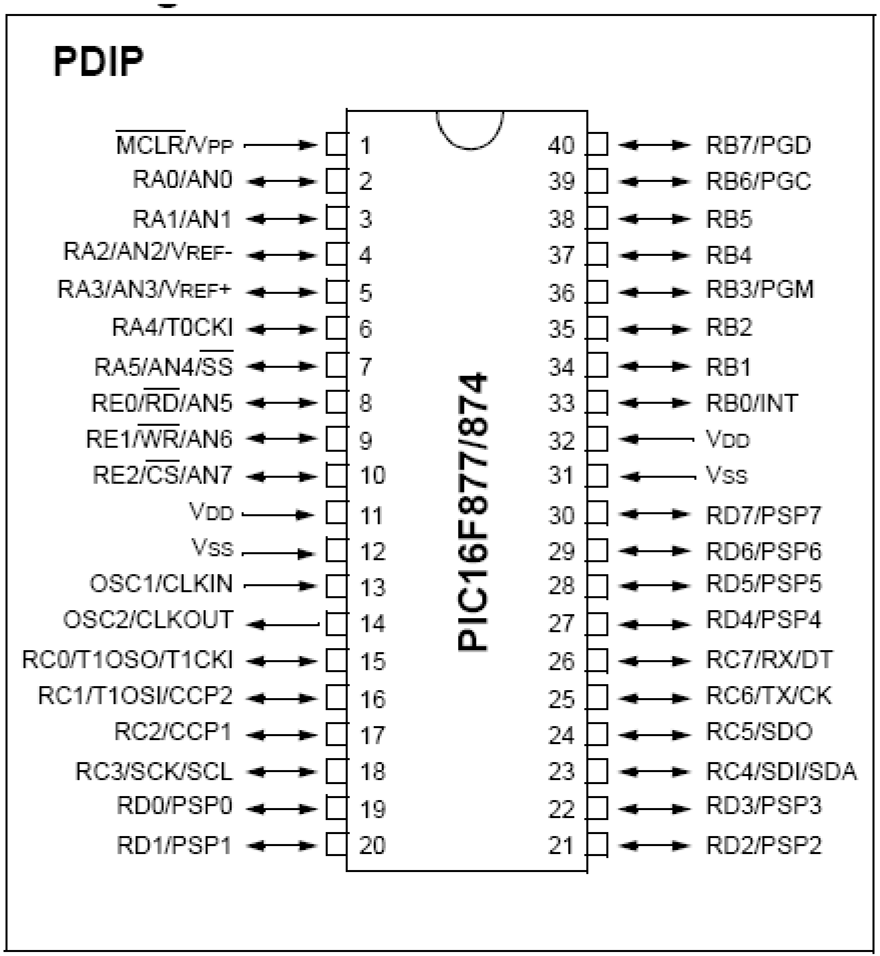


Figure 2.5: Pin diagram

Referring to Figure 2.5, it shows each pin has its meaning. But I only want to justify for the common pins where it will be function. For pin no. 1 (V_{PP}), pin no. 11 (V_{DD}), pin no. 32 (V_{DD}), these pins will be connecting to the 5V voltage. For pin no. 12 (V_{SS}), pin no. 31 (V_{SS}), these pins will be connecting to the ground (GND). For pin no. 13 (OSC1) and pin no. 14 (OSC2), these pins will be connecting to the oscillator. For other pins, we take point for the input and output. As example, for pin no. 2 (RA0/AN0), this pin is zero pin on port A. For pin no. 33 (RB0/INT), this pin is zero on port B. We can select or program to give command either port A or port B to be input or output.

The most important part is how to program this PIC16877 microcontroller. For this project, the PICBasic is the preferred compiler to build program for the microcontroller. For PIC microcontroller, we found that we can use two options that are assembly language and PicBasic programming. It is easier and quicker if we use PicBasic programming than the use of assembly language programming. The English Basic language is much easier to read and write down quickly than microchip assembly language. PPICBASIC is a micro-controller developed by COMFILE Technology and is programmed in a form of the BASIC language. Mostly, PICBasic use the basic language such as MOVE, LOW, HIGH, GOTO, GOSUB and others. PICBASIC has BASIC interpreter inside of a microcontroller on board so that you can program micro controller in a form of the BASIC language with ease. We will program the microcontroller by copy the command to the PIC microcontroller (EEPROM).

2.4 ROBOTICS.

Robots are very powerful element of today's industries. They are capable of performing many different task and operation precisely and do not require common safety and comfort element human need.

Even so robots are powerful element there are some advantages and disadvantages of the robot as shown in Table 2.1.

Table 2.1: The advantages and disadvantages of robots.

Advantages	Disadvantages
Increase productivity, safety, efficiency, quality and consistence of product.	Robots lack capability to respond in emergency, unless the situation is predicted and the response is included in the system.
Work in hazardous environment without the need of life support, comfort, or concern about safety.	Have limited capability in degree of freedom, dexterity, sensor, vision system and real time response.
Robots need no environment comfort, such as lighting, air conditioning, ventilation, and noise protection.	Robots are costly due to initial cost of equipment, installation cost, need for peripherals, need for training and need for programming.
Robots work continuously without experiencing fatigue or boredom, do not get mad and need no medication insurance or vacation.	
Robots have repeatable precision at all the time, unless something happen to them or unless they wear out.	

2.4.1 ROBOT COMPONENT.

Generally robot as a system does consists of the following elements as show in Table 2.2 below, which are integrated together to form a whole robot perfectly.

Table 2.2: Elements of robots.

Elements.	**Explanation.**
Manipulator, or rover	Main body of the robot and consist of the link, the joints and other structural element of the robot.
End effecter	Connected to the last joint (hand) of the manipulator, which generally handle the object or connected to machine to perform task also called tools.
Actuator	The "muscle" of the manipulator, commonly type such as servomotor, stepper motor, pneumatic cylinder and hydraulic cylinder.
Sensors	To collect information about the internal state of the robot or to communicate with the outside environment.
Controller	Rather similar to our cerebellum it's to control the motion of the robot joint and tools.
Processor	Is the brain of the robot, it calculates the motions of the robot joints, determine how fast or how far the joint must move.
Software	Is the learning program for the robot which is determined by user

2.4.2 ROBOT KINEMATICS: POSITION ANALYSIS.

The forward kinematics will enable us to determine where the robot's end (hand) will be if all joint variable are known. Inverse kinematics will enable us to calculate what each joint must be if we desire that the hand be located at a particular point and have particular orientation.

In this analysis matrices will be used to represent points, vectors, frames, translation, rotations, and the transformation, as well as object and other kinematic elements in a frame.

Representation of a pure Translation:

$$Trans(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{------------------------------- (1)}$$

Representation of a pure Rotation about an Axis:

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{-------------------------------- (2)}$$

$$Rot(y,\theta) = \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{--------------------------------} \ (3)$$

$$Rot(z,\theta) = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{--------------------------------} \ (4)$$

2.4.3 DENAVIT- HARTENBERG REPRESENTATION OF FORWARD KINEMATIC EQUATION OF ROBOTS.

This Denavit-Hartenberg (D-H) technique has become standard way of representing robots and modeling motion. Is a very simple way of robot link and joints that can be used for any robot configuration regardless of its sequence and complexity.

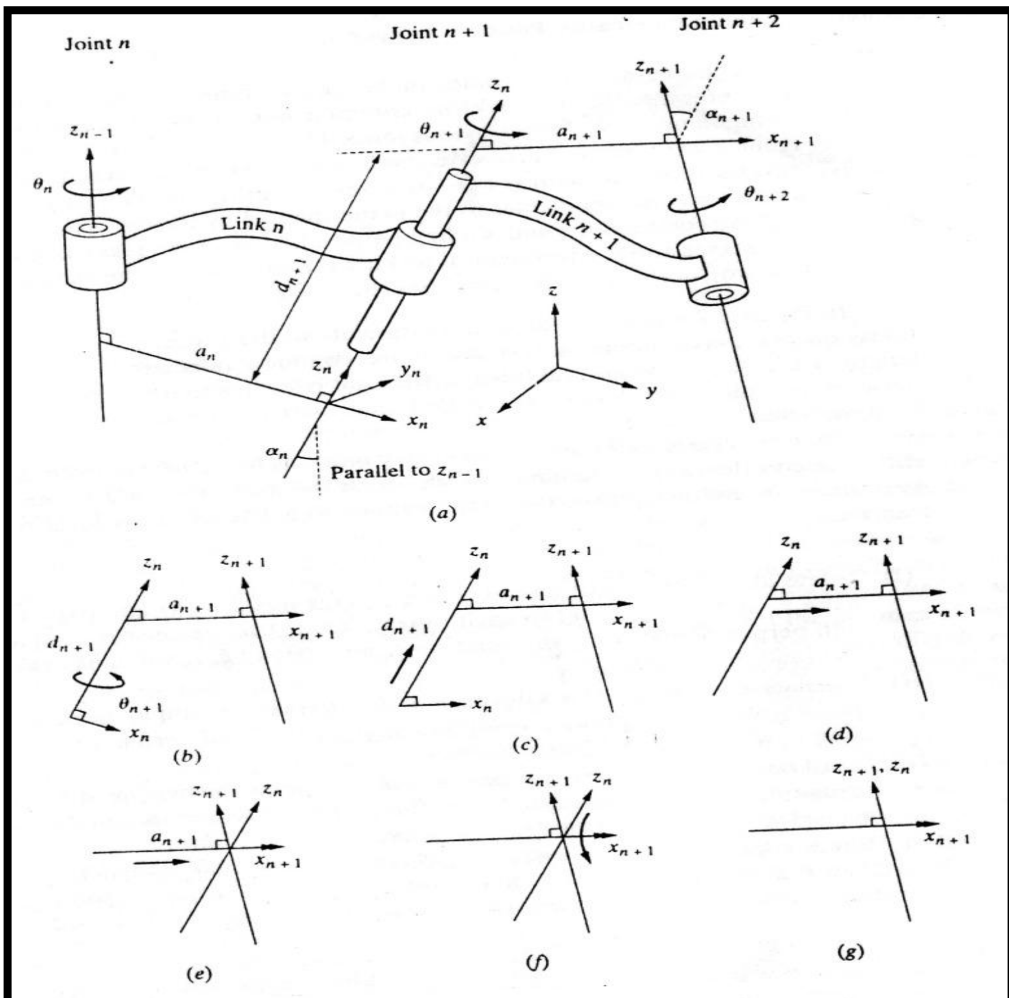


Figure 2.6: A D-H representation of a general purpose joint link combination.

(I) Rotate about the z_n –axis an angle of θ_{n+1} (Figure 2.6a and 2.6b). this make x_n and x_{n+1} parallel each other. This is true because a_n and a_{n+1} are both perpendicular to z_n and rotating z_n angle of θ_{n+1} will make them parallel.

(II) Translate along z_n –axis a distance of d_{n+1} to make x_n and x_{n+1} collinear (Figure 2.6c). Since x_n and x_{n+1} were already parallel and normal to z_n, moving along z_n will lay them over each other.

(III) Translate along the x_n –axis a distance of α_{n+1} to bring origins of x_n and x_{n+1} together (Figure 2.6d and 2.6e). At this point, the two origin of the two reference frame will be at the same location.

(IV) Rotate z_n –axis about x_{n+1} –axis an angle of α_{n+1} to align z_n –axis with z_{n+1} –axis (Figure 2.6f). At this point, frames n and n+1 will be exactly the same (Figure 2.6g), and we have transformed from one frame to next.

Doing exactly the same sequence of four movement between n+1 and N+2 frame will transform one to the next, and by repeating this as necessary, transform between successive frame.

$$^{n}T_{n+1} = A_{n+1}$$

$$= Rot(z, \theta_{n+1}) \times Trans(0,0, d_{n+1}) \times Trans\ (a_{n+1}, 0,0) \times Rot(x, \alpha_{n+1}) \text{ --- } (1)$$

$$= \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1} & 0 & 0 \\ S\theta_{n+1} & C\theta_{n+1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & a_{n+1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_{n+1} & -S\alpha_{n+1} & 0 \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ --------------------- (2)}$$

$$A_{n+1} = \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1}C\alpha_{n+1} & S\theta_{n+1}S\alpha_{n+1} & a_{n+1}C\theta_{n+1} \\ S\theta_{n+1} & C\theta_{n+1}C\alpha_{n+1} & -C\theta_{n+1}S\alpha_{n+1} & a_{n+1}S\theta_{n+1} \\ 0 & S\alpha_{n+1} & C\alpha_3 & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ -------------------- (3)}$$

Where *n* is the joint number.

To facilitate the calculation of A matrices, Table 2.3 can be used to represent value each link and joint are determined from the schematic drawing of the robot.

Table 2.3: D-H Parameters table.

#	**θ**	***d***	***a***	**A**
1				
2				
3				
4				
5				
…				

2.5 SENSORS.

Sensor is used for both internal feedback control and external interaction with the outside of the environment. There are many different type of sensor available that measure deferent phenomena.

To choose an appropriate sensor for a particular need, one has to consider a number of different characteristics. These characteristic determine the performance, economy, ease of application, and applicability of the sensor. The characteristic of the sensor that are needed to be considered are shown in Table 2.4.

Table 2.4: Sensor characteristic that need to be consider.

Characteristics.	Explanations.
Cost	Cost must be balanced with requirements of the design, such as reliability, importance of the data provide, accuracy and life.
Size	Is important to ensure enough room for the sensor in the system.
Weight	Since robots are dynamic machine, the weight of the sensors is very important.
Type of output	The output of sensor maybe analog or digital, and depending on the application, this output may be used directly or need to be converted.
Interfacing	Sensor must be interface with another device such as controller and microprocessor. If do not match add-on circuit are necessary.
Resolution	Resolution is the minimum step size within the range of the sensor. $Resolution = \frac{full\ range}{2^n}$
Sensitivity	Is the ratio of a change in output in response to a change in input.
Linearity	The same change in input at any level within the range will produce the same change in output.
Range	Is the different between the smallest and the largest outputs the sensor can produce, or the difference between the smallest and the largest inputs with which it can operate properly.
Response time	Time required to observe the change in outputs as the result the change of in inputs.
Frequency response	Is the range of the frequency response, the better ability of the system to respond to varying input.
Reliability	The ratio of how many times system operates properly divided how many times it is tried.
Accuracy	How close the output of the sensor is to the expected value.
Repeatability	How varied the different outputs are relative to each other.

2.5.1 PROXIMITY SENSORS.

Proximity sensors are used to determine that an object is close to another object before contact is made. This noncontact sensing can be useful in many situations, from measuring the speed of the rotor, to navigating the robot. There are many different types of proximity sensors, such as magnetic, eddy current, Hall- effect, optical, ultrasonic, inductive, and capacitive.

2.5.2 OPTICAL PROXIMITY SENSORS.

Optical sensors consist of light source called emitter (either internal or external to the sensors) and a receiver, which sense the presence or the absence of light. The receiver usually a phototransistor and the emitter is usually an LED. The combination of the two creates a light sensor.

As a proximity sensor, the sensor is set up such that the light, emitted by the emitter, is not received by the receiver, unless an object is close- by. Figure 2.7 is a schematic drawing of optical proximity sensor. Unless a reflective object is within range of the switch, the light is not seen by the receiver, and, therefore, there will be no signal.

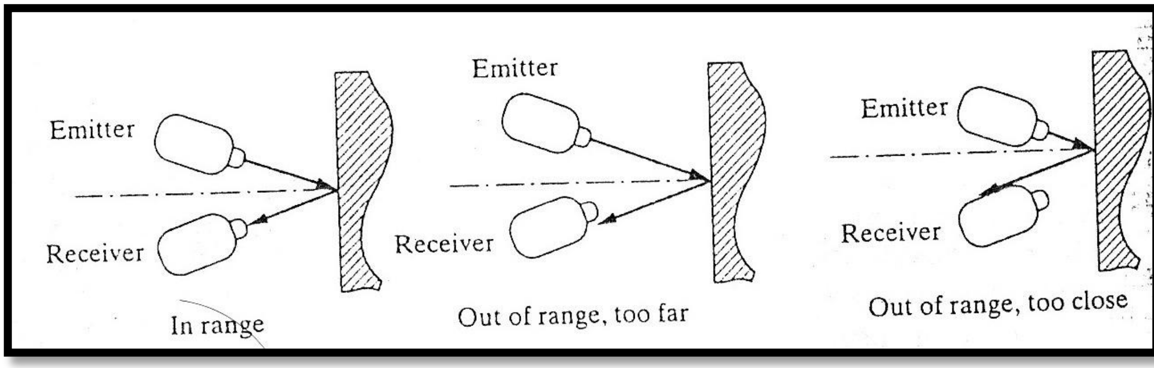


Figure 2.7: Optical proximity sensors.

2.5.3 INFRARED (IR) SENSORS.

Infrared (IR) is the typical light source used as a sensor in hobby robot to sense opaque object. The basic principle of IR sensor is based on an IR emitter and an IR receiver. IR emitter will emit infrared continuously when provide power source to it. Since there is no source of power for IR receiver, it would not emit any light. It will only receive infrared if there is any. Usually we will attach the IR emitter and IR receiver side by side, and point them to a reflective surface. The advantage of IR sensor compare to other optic sensor is ideal sensor during daylight condition, cheap and easy to construct. Figure 2.8 show the diagram of IR reflection to the object.

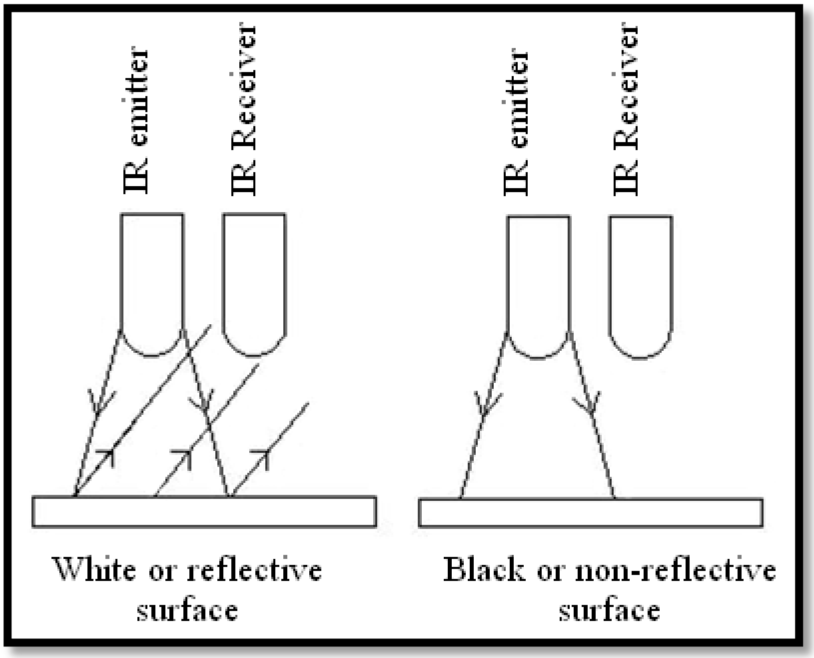


Figure 2.8: IR sensor reflections.

2.5.4 ULTRASONIC PROXIMITY SENSORS.

In these sensors, an ultrasonic emitter emits frequent burst of high- frequency sound wave around 200kHz range which is can't be heard by humans. There are two modes of operation which is opposed mode and echo (diffused) mode. In opposed mode, the receiver replace in front of the emitter, whereas an echo mode, the receiver is either next or integrated with into, the emitter and receiver the reflected sound wave. If the receiver is within range or if the sound is reflected by a surface close to the sensor, it will sense and a signal will produced. Otherwise, the receiver will not sense wave, and there are no signal produced. Figure 2.9 is a schematic drawing of this type of sensor. The disadvantage of this sensor is expensive and not easy to implement.

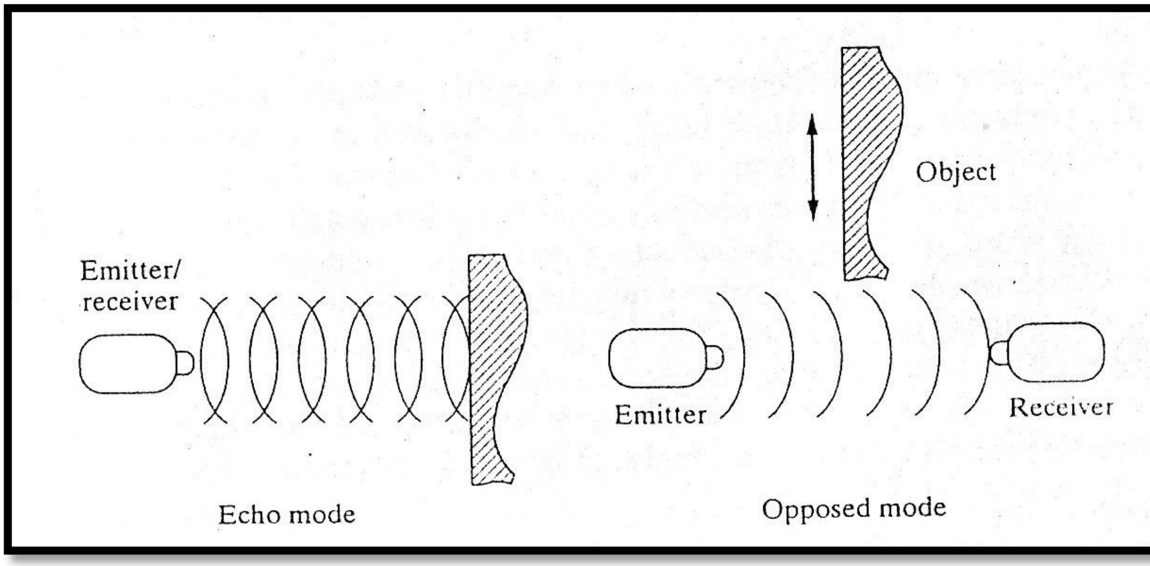


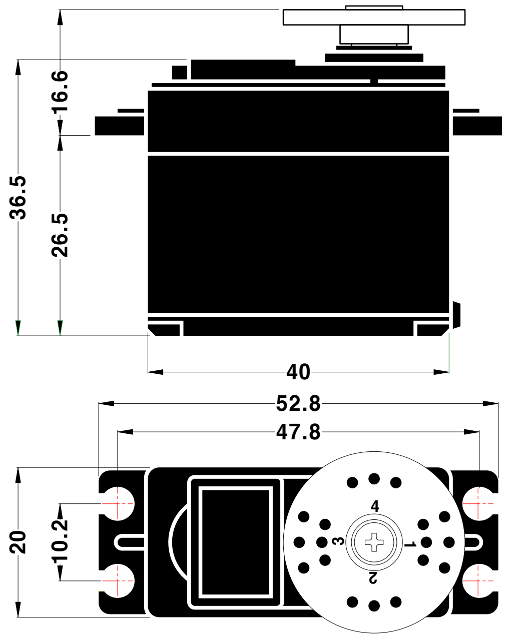
Figure 2.9: Schematic ultrasonic sensors.

PREPARED BY JUN HEE, LEE
UPDATE:APR 01, 2002

ANNOUNCED SPECIFICATION OF HS-322HD STANDARD DELUXE SERVO

1.TECHNICAL VALUE

CONTROL SYSTEM	:+PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	:4.8V TO 6.0V	
TEST VOLTAGE	:AT 4.8V	AT 6.0V
OPERATING SPEED	:0.19sec/60? AT NO LOAD	0.15sec/60? AT NO LOAD
STALL TORQUE	:3kg.cm(41.66oz.in)	3.7kg.cm(51.38oz.in)
IDLE CURRENT	:7.4mA AT STOPPED	7.7mA AT STOPPED
RUNNING CURRENT	:160mA/60? AT NO LOAD	180mA/60? AT NO LOAD
STALL CURRENT	:700mA	800mA
DEAD BAND WIDTH	:5usec	5usec
OPERATING TRAVEL	:40? /ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	:CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
MOTOR TYPE	:CORED METAL BRUSH	
POTENTIOMETER TYPE	:4 SLIDER/DIRECT DRIVE	
AMPLIFIER TYPE	:ANALOG CONTROLLER & TRANSISTOR DRIVER	
DIMENSIONS	:40x20x36.5mm(1.57x0.78x1.43in)	
WEIGHT	:43g(1.51oz)	
BALL BEARING	:TOP/RESIN BUSHING	
GEAR MATERIAL	:2 HEAVY DUTY RESIN	
HORN GEAR SPLINE	:24 SEGMENTS/?5.76	
SPLINED HORNS	:RESULAR/R-C,R-D,R-I,R-O,R-X,SUPER/R-XA	
CONNECTOR WIRE LENGTH	:300mm(11.81in)	
CONNECTOR WIRE STRAND COUNTER	:40EA	
CONNECTOR WIRE GAUGE	:25AWG	



2.FEATURES

LONG LIFE POTENTIOMETER, TOP RESIN BUSHING, 2 HEAVY DUTY RESIN GEARS

3.APPLICATIONS

AIRCRAFT 20-40 SIZE,STEERING AND THROTTLE SERVO FOR CARS, TRUCK AND BOATS

4.ACCESSORY & OPTION

CASE SET/
- HS322T:1EA
- HS322M:1EA
- HS322L:1EA
- PH/T-2 2x30 NI:4EA

GEAR SET/
- HS322G1:1EA
- HS325G2:1EA
- HS325G3:1EA
- HS322G4:1EA
- HS300RB:1EA

HORN SET/
- R-C:1EA
- R-D:1EA
- R-I:1EA
- R-O:1EA
- R-X:1EA
- R-XA:1EA
- WH/W 2.1x15 NI:4EA
- BST 3x5.5:4EA
- NBR 9x6.5x6:4EA

HITEC RCD KOREA INC.

CHAPTER 3

SYSTEM DESIGN

3.1 INTRODUCTION.

This whole project basically is about controlling the robot consist of servos at the leg that controlled by the controller and sensing using sensor to avoid the obstacle. A PIC16F877A microcontroller is been used to control and manipulate all 8 servo motors that are attached to the robot leg. This is been done so that the controlling of walking straight, turning left movement could be performed. The flow chart in Figure 3.0 below shows the system of robot's movement.

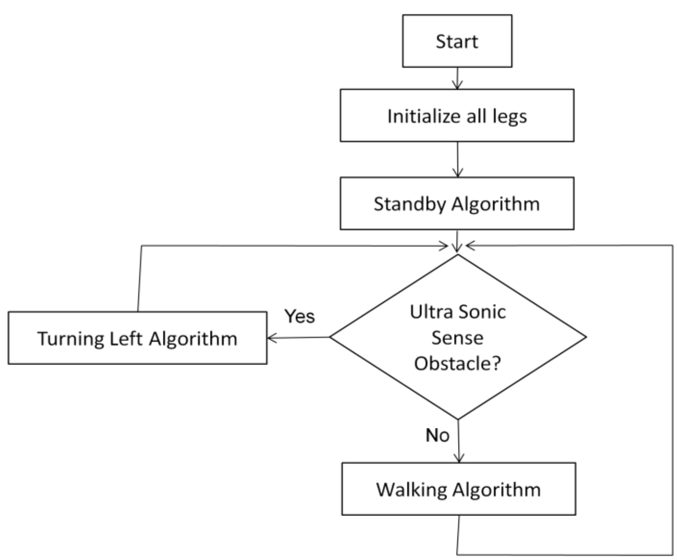


Figure 3.0: Flow chart of the robot system.

PIC controller controls the entire servo based on the crawling algorithm design based on animal movement in stable fashion so that the robot will be able to walk ahead, turn left and able to avoid obstacle, by sending the PWM to each servo so it performs the desired movement through programming basis.

Figure 3.1 shows the picture of the quadruped robot. In this project it divided into two parts which is software and hardware implementation. Each of the part will be discuss in the following chapter ahead.

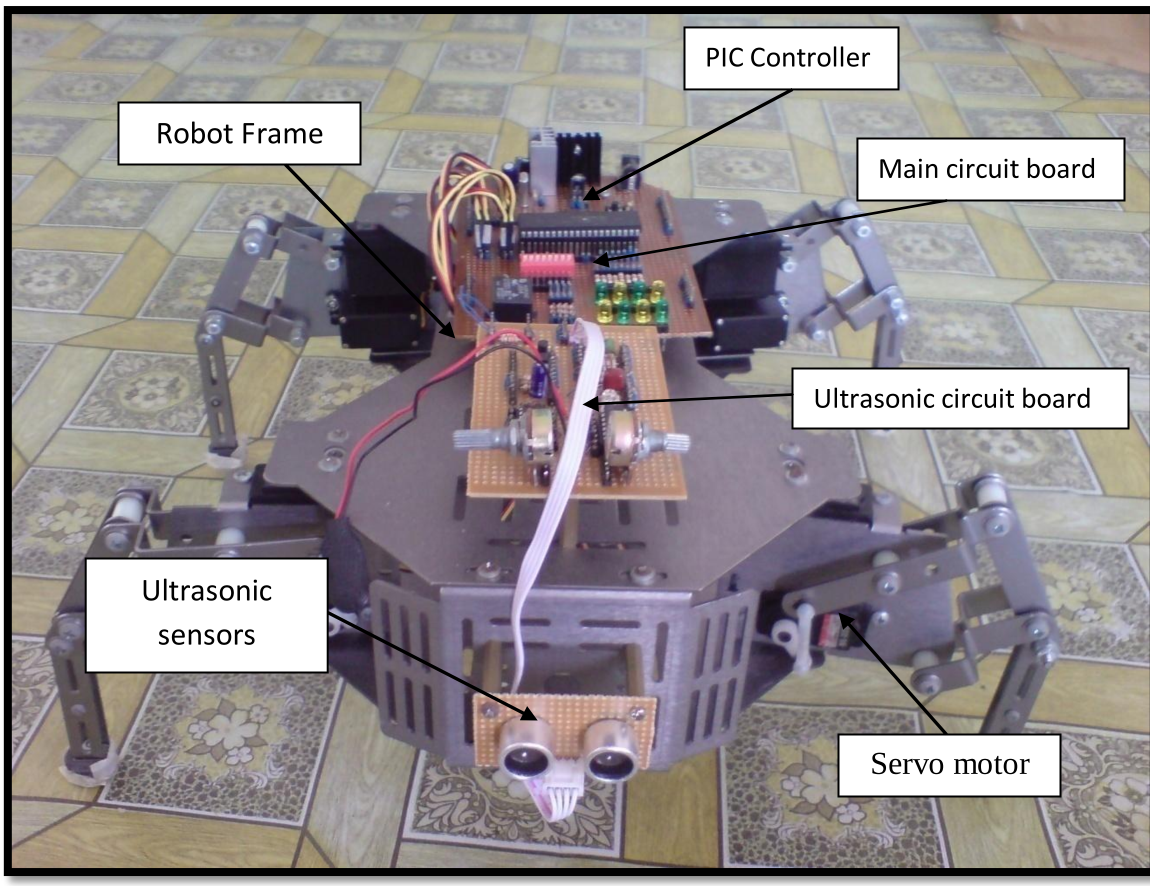


Figure 3.1: Picture of the robot.

3.2 HARDWARE IMPLEMENTATION.

Throughout this project, few modules and devices have been developed. It consists of power supply, actuator, controller, and sensing module.

3.2.1 POWER SUPPLY MODULE.

This module is based on supplying the project Microchip PIC16F877A microcontroller and the servo motor +5V voltage power supply to operate. There for we need to build regulated +5V source. Usually the supplied voltage before regulation will be ranging from 9V to 24V DC (direct current). But recommended voltage is 6V to 12V only. Higher the voltage might damage the regulator.

To make a +5V power supply, LM7805 voltage regulator IC (integrated circuit) was used. The IC show in Figure 3.2.

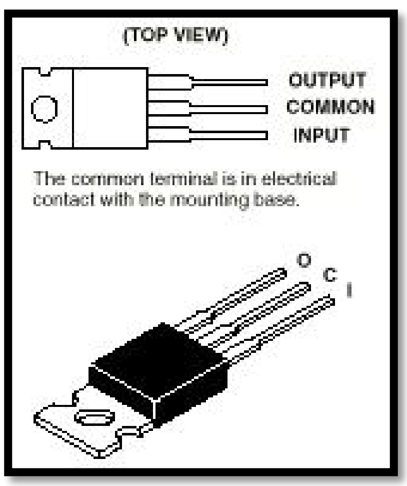


Figure 3.2: LM7805 regulator.

Sometimes the unregulated voltage input might have some distortion and it results the output do consists distortion and unstable +5V. In order to smooth out and to get better +5V output voltage capacitor is been used. The Figure 3.3 below shows the circuit of +5V power supply that been developed.

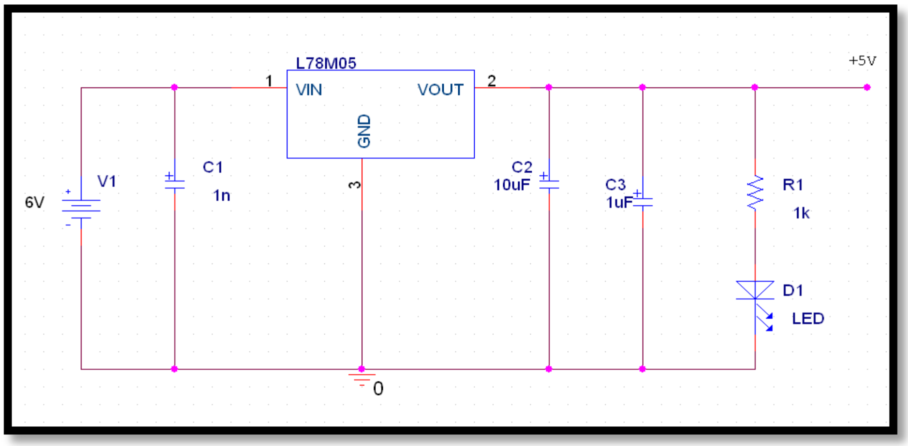


Figure 3.3: Schematic circuit of +5V power supply.

3.2.2 PIC MICROCONTROLLER.

This chapter doesn't only covers about controller alone, yet the programmer device, oscillator circuitry, and the connection that been made for the LED module and the connection for the all 8 servo motors.

3.2.3.1 PIC

The function of this microcontroller is as a brain to the robot that controls the entire servos so it could move according to the movement algorithm that being set throughout programming. Microchip PIC16F877A microcontroller has been chosen for this project. This microcontroller is been chosen because PIC16F877A controller is an advance controller compared normal Motorola MC68HC11A1 controller. It also is a low cost and a commonly used controller.

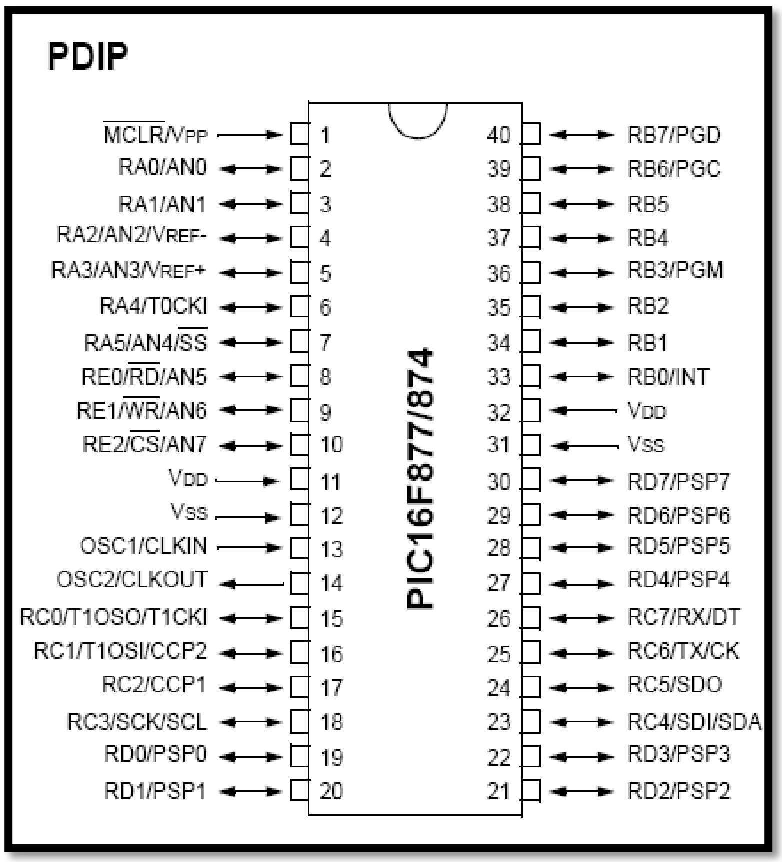


Figure 3.4: Pin configuration for PIC16F877A

3.2.3.2 OSCILLATOR AND RESET

The OSC1 or CLKIN (pin 13) and OSC2 or CLKOUT (pin 14) used to provide the internal clock. The CLKOUT pin does produce constant high output and it goes through the crystal. From the crystal, it will form the desired square waveform as the crystal circuit value 8MHz and finally enters back to CLKIN pin. And the pulse will provide clocking for the PIC microcontroller to be used throughout the whole project. The reset configuration is used to reset the whole microcontroller. The circuit is directly from +5V goes through 10KΩ and through a switch. Active low is used for triggering the reset for the controller.

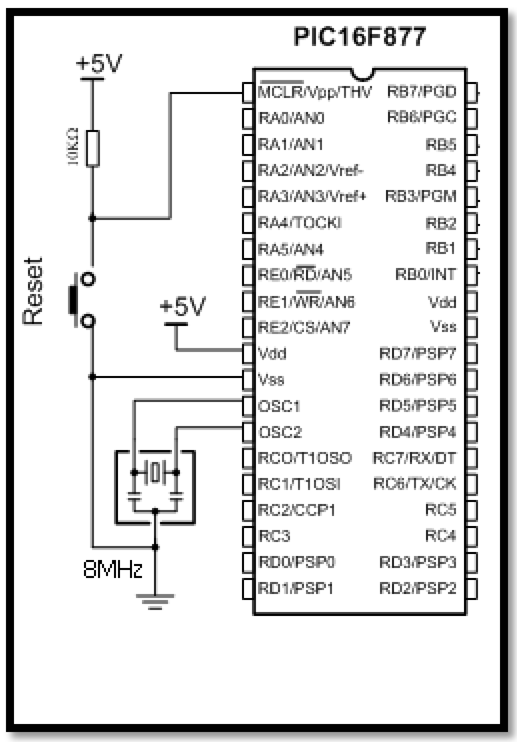


Figure 3.5: Oscillator circuit.

3.2.3.2 PROGRAMMER

Every program that we design need to be programmed to the controller so that it could operate as we desired. In this project, UIC00A USB ICSP PIC PROGRAMMER has been used. It is been attached to a header in the main board of microcontroller and wrapped to the PIC16F877A. The Figure 3.6 shows the picture of the programmer and Figure 3.7 shows the diagram how to integrate with the controller.

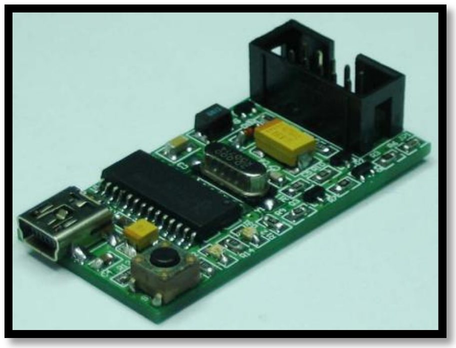


Figure 3.6: USB ICSP PIC PROGRAMMER

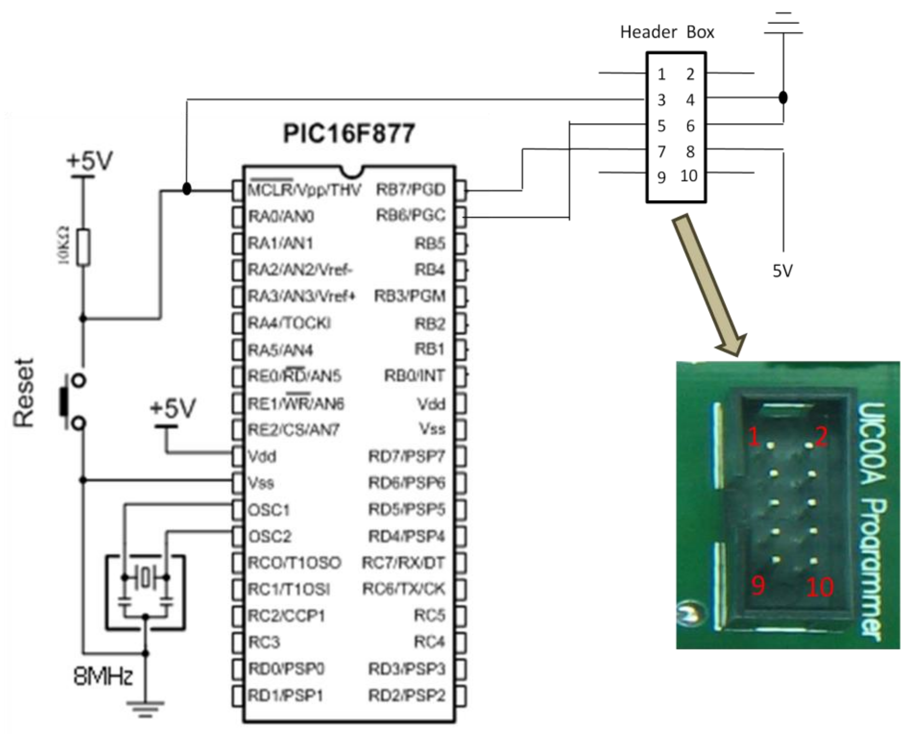


Figure 3.7: Connection at main development board

3.2.3.3 LED CONNECTION

These LED's are been implemented in this project for the study purpose. Since the high level language of PICBasic Pro is a new language, this circuit is been design to learn how the high level language actually works in this PIC16F877A. How to control all the LED's according to the program that we develop. Below Figure 3.8 shows the configuration that been used for the LED. All the LED is been connected to PortB that names RB0 until RB7 and resistor valued 330Ω is been connected series to ensure the current that passes through the LED is controlled and doesn't burst the LED itself.

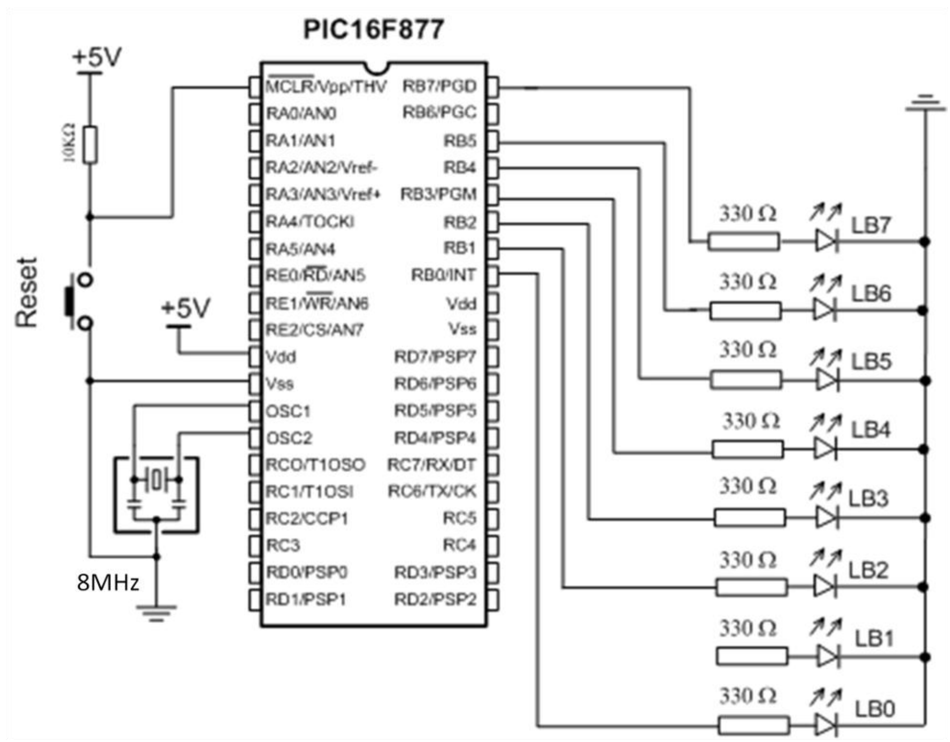


Figure 3.8: LED connections at PortB

3.2.4 SERVO MOTOR

These servo motors are been used throughout this whole project. The 4 leg robot was widely used in legged robots. The degree of freedom for the 4 legged robot is determined on the how many of joint the robot consist at each leg, since this robot have 2 joint at each leg there for the robot have 2 degree of freedom. Two servo motors here are attached to the leg of the robot means each leg 2 motor. The leg frame of the robot is shown clearly in diagram and a picture in Figure 3.9. One servo gives a degree of freedom at a shoulder position and another servo gives degree of freedom at the arm of the robot.

The type of joint for this robot, is rotational joint type which is this type provide a rotational relative motion of the joint, with the axis of the rotation perpendicular to the axes of the input and output link.

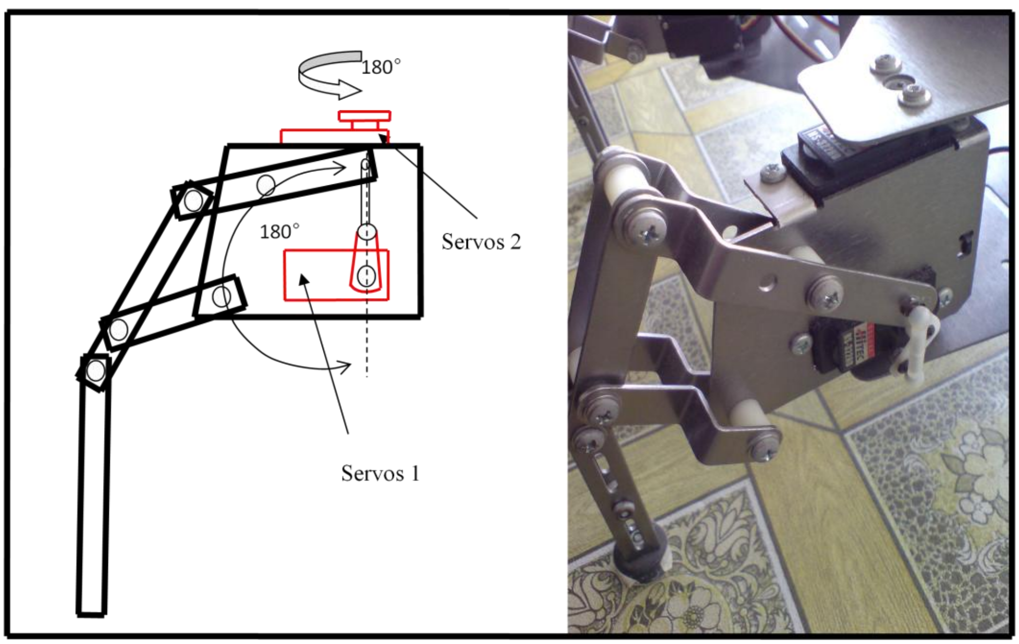


Figure 3.9: Position of the servo motor in the leg.

Figure 3.10 shows the connection that has been made from controller to servo motor. From the diagram, we can see that connection is at PortC microcontroller. Table 3.0 shows the connections for the motors are made according to the positioning of the robot:

Table 3.0: Port connection to the leg

Right Front Shoulder	PortC.0
Right Front Arm	PortC.1
Right Behind Shoulder	PortC.2
Right Behind Arm	PortC.3
Left Behind Arm	PortC.4
Left Behind Shoulder	PortC.5
Left Front Arm	PortC.6
Left Front Shoulder	PortC.7

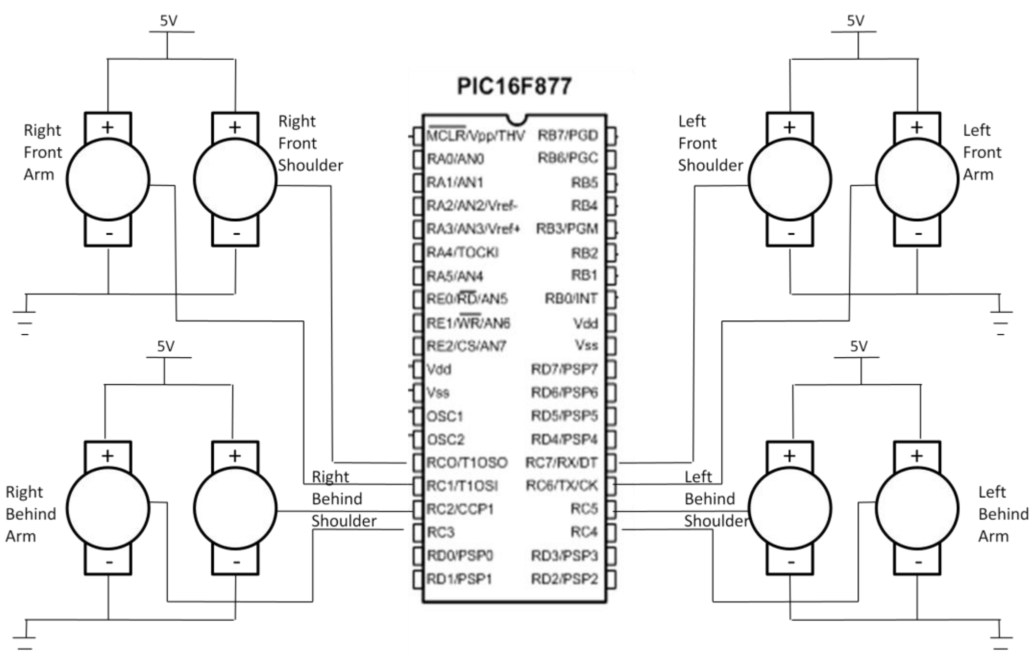


Figure 3.10: Circuit Configuration for all 8 Servo Motor

3.2.5 ULTRASONIC SENSOR

For the avoidance module that been implemented in this project is ultrasonic sensor. Functions to detect the obstacle in front and tells microcontroller about it. This sensor is been selected because of several reasons:

i. It's a way too efficient sensor compared infrared sensor.
ii. The sensor can be used in daylight.
iii. It could detect the obstacle in longer distance.

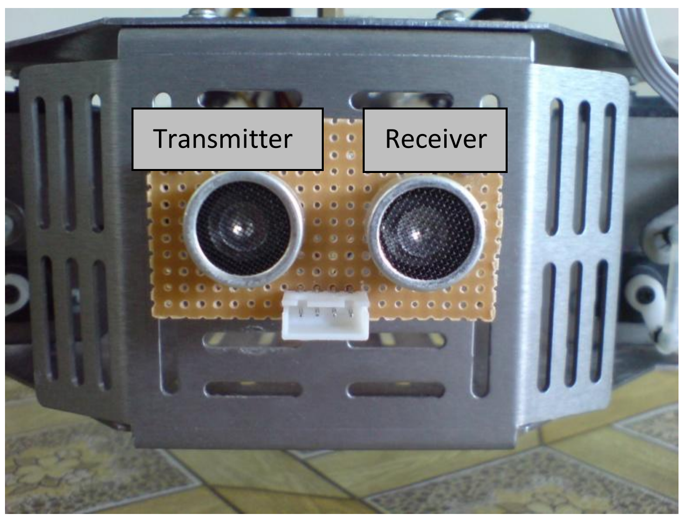


Figure 3.11: Ultrasonic sensor at the quadruped robot (front view).

Figure 3.11 shown above, it is implementation of an ultrasonic sensor. It acts as an operator's vision and intelligence for guidance and navigation. The sensor that been implemented is based on single transmitter and single receiver only. Attaching the ultrasonic does had a problem that is it produces only 2V as an output. So, internal ADC in the PIC is implemented. Finally the 2V as an output could be received successfully.

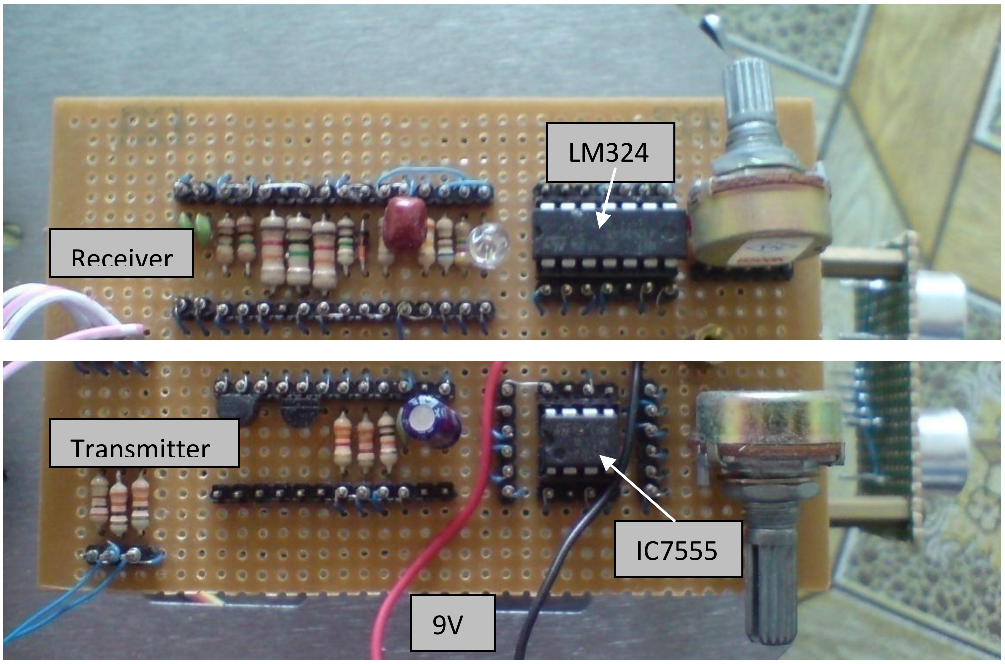


Figure 3.12: The circuit design for ultrasonic (upper view).

Picture above Figure 3.12 shows the circuit for the ultrasonic sensor. This ultrasonic proximity detector comprising independent, battery 9V powered transmitter and receiver sections makes use of a pair of matched ultrasonic piezoceramic transducers operating at around 40 kHz each. Figure 3.13 shows the transmitter circuit.

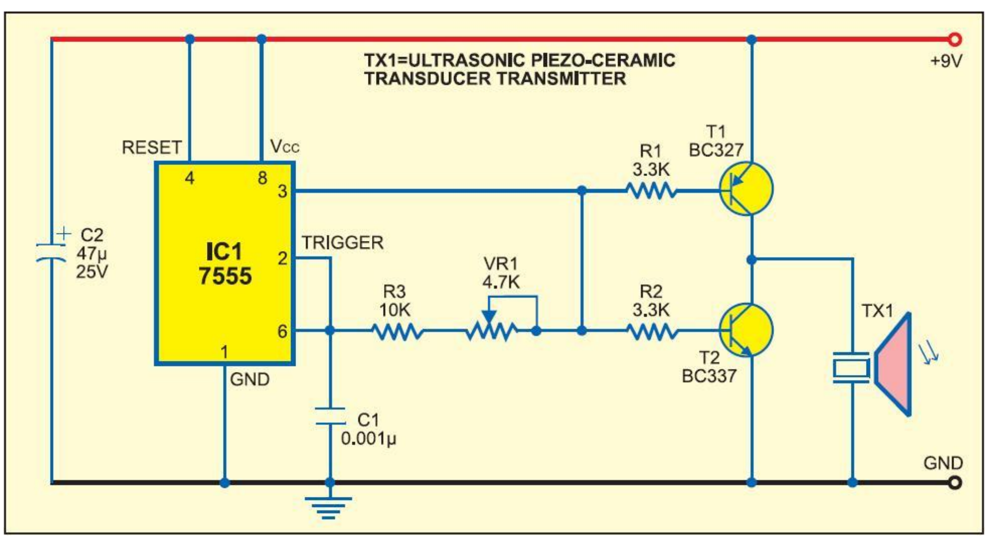


Figure 3.13: Transmitter circuit

It comprises CMOS timer IC 7555 (IC1) configured as an astable multivibrator, which may be tuned to the frequency of the ultrasonic piezoceramic transmitter's resonant frequency of around 40 kHz using preset VR1. A complementary pair of transistors T1 and T2 is used for driving and buffering the transducer while it draws spikes of current from IC1 circuit to sustain oscillations and thereby avoids any damage. The receiver front-end (refer Figure 3.14) is designed to provide a very high gain for the reflected faint ultrasonic frequency signals detected by the ultrasonic transducer. The amplifiers built around N1 and N2 respectively, provide AC voltage gain of around 80 each. These two stages should have a high open-circuit gain, product, it signifies his interest. Switching diode D1 followed by a filter comprising capacitor C5 and resistor R10 is used to meet this requirement. The filter also helps to bypass brief bursts of ambient noise in the ultrasonic wide bandwidth and very low bias current apart from being capable of single-supply operation.

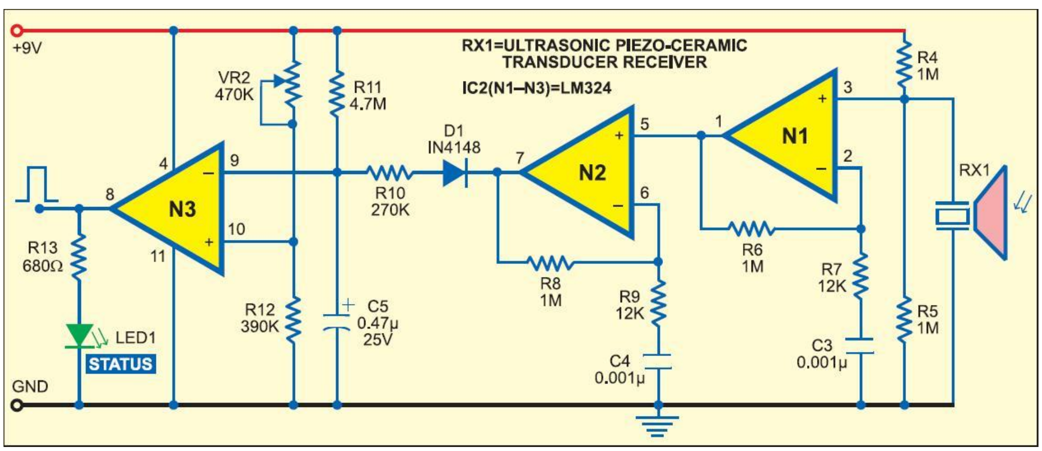


Figure 3.14: Receiver circuit.

Quad op-amp LM324 is used here due to its low cost. For higher efficiency, you may use single op-amps such as CA3130 or CA3140. When a visitor auses before a range. The third stage comprising N3 works as a comparator to provide a triggering pulse when a visitor stops by. This pulse can be used to trigger a timer or a monostable, whose output may then be used to switch on the audio/video message concerning the product for a predetermined period. When somebody comes in front of the ultrasonic piezoceramic transducer pair, the status LED (LED1) glows because of the signal reflected from the body of the visitor. The circuit can be assembled on any general-purpose PCB. The transmitter and the receiver should be aligned such that the transmitted ultrasonic signal is optimally received by the receiver after reflection. Installation of the ultrasonic piezoceramic transducer pair operates at around 40 kHz.

3.2.6 THE ROBOT MOVEMENT DESIGN.

The movement of this robot was design carefully so it can walk in stable fashion without falling during the movement. The movement of the robot was design based on the animals' movement as the reference and guidance. The animal that take as the reference is frog, the frog we taken because the characteristic of the frame of the robot is almost the same as the frog, the robot have four leg and it same as the frog also have four leg. Also, the leg position of the robot is almost the same as the frog, but the leg of the robot is totally different from the frog. Below is the movement of frog that has been analysis to design a robot movement to walk straight.

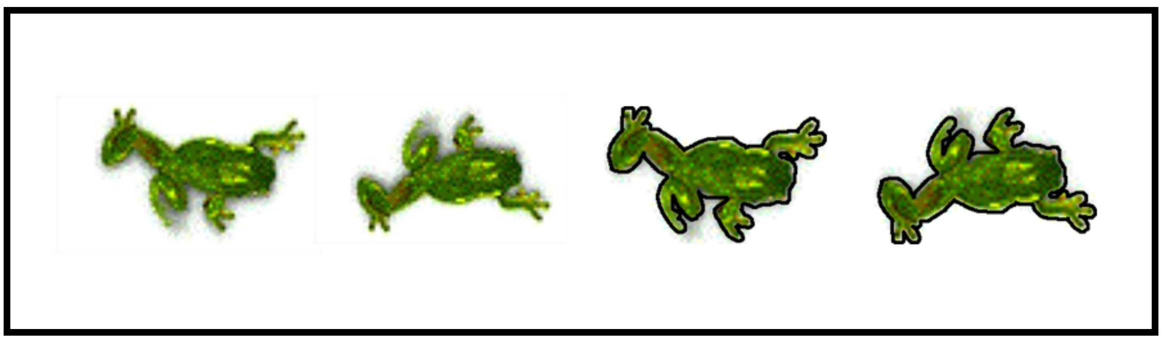
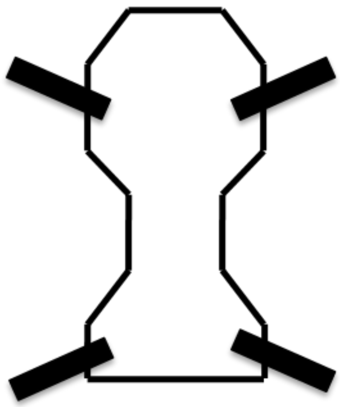


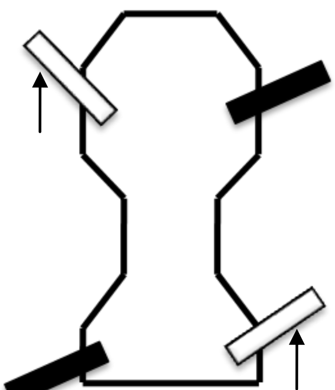
Figure 3.15: Frog Movement.

Figure 3.15 shows the design of robot movement to walk straight based on the frog movement analysis, the movement of the robot design is not 100% of the frog movement it is because to ensure the robot always stable during the walking mechanism, but still these movement shown in picture is considered for the algorithm development, since these are the main step to ensure the robot could move front efficiently. The black color indicate the leg are on the ground, the white show the leg are rise up and the arrow show the movement of leg.

3.2.6.1 WALKING ALGORTIHM

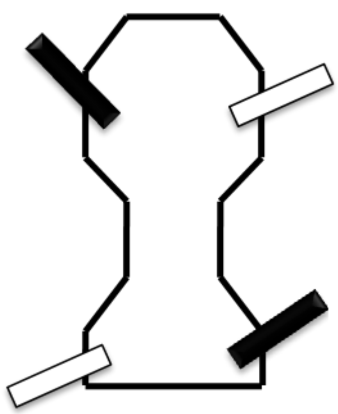


(1) Robot at rest, all legs landed on the ground.



(2) Left front leg arise from previous condition and goes to new position.

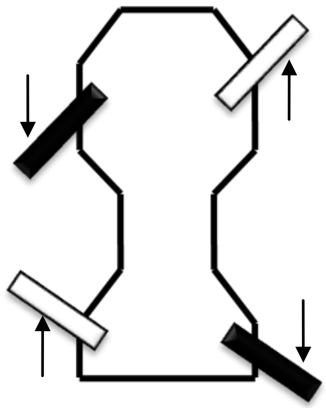
(3) Right behind leg arise from previous condition and goes to new position.



(4) Left front leg and right behind leg falls to ground.

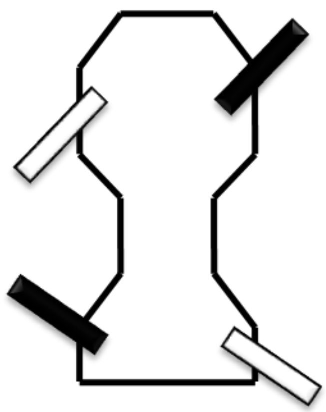
(5) Right front leg and left behind leg arise.

Figure 3.16: The movement style for robot walking straight.



(6) The left front leg and right behind leg goes behind.

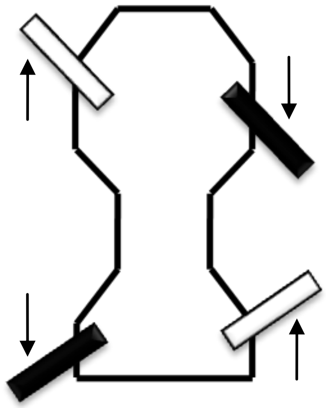
(7) The right front leg and left behind leg goes front.



(8) Left front leg and right behind leg arise.

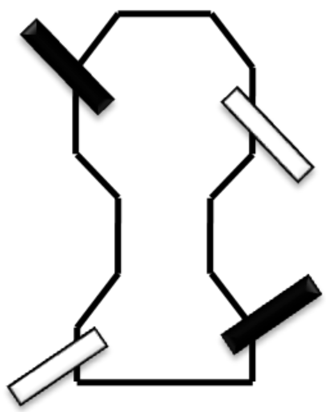
(9) Right front leg and left behind leg falls to ground.

Figure 3.17: The movement style for robot walking straight (cont).



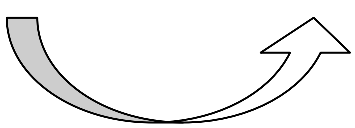
(10) The left front leg and right behind leg goes in front.

(11) The right front leg and left behind leg goes behind.



(12) Left front leg and right behind leg falls to ground.

(13) Right front leg and left behind leg arises.

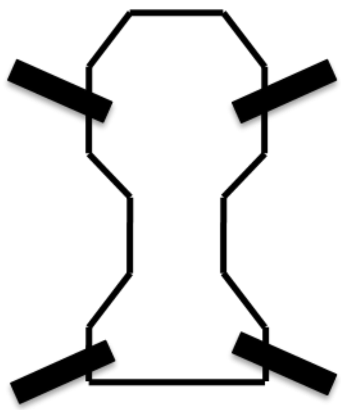


(14) Repeat the module from step (6)

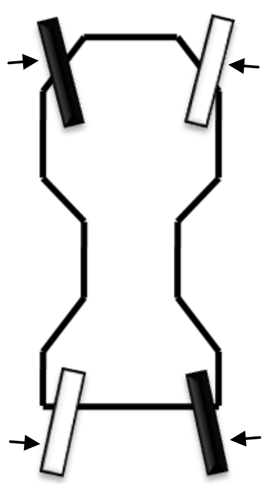
Figure 3.18: The movement style for robot walking straight (cont).

3.2.6.2 TURNING LEFT ALGORITHM

For the movement to robot make a turn left, the design were done without based on animals, this is because to make the robot turn is totally complex compare the robot walking straight, there for the movement of robot to make turn left was design 100% without refer to the frog movement. Below show how the robot turn left.



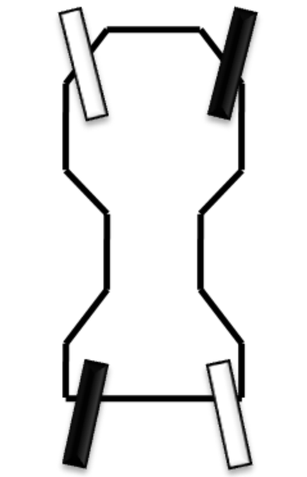
(1) Assume all the leg is at initial condition rest in ground.



(2) Right front leg and left behind leg arises.

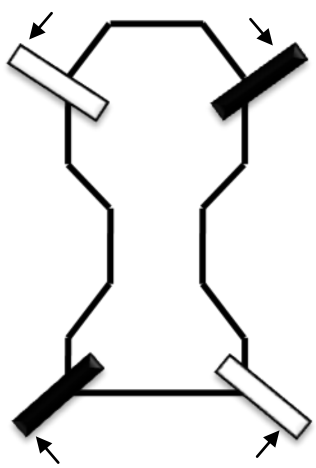
(3) All the legs in their position of up and down nears inside body.

Figure 3.19: The movement style for robot to turn left.

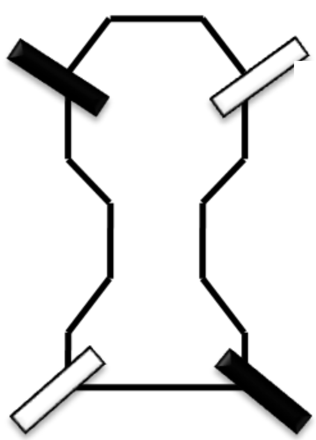


(4) Right front leg and left behind leg lands down to ground.

(5) Left front leg and right behind leg arises.



(6) All the legs in their position of up and down goes far away from the body.



(6) Right front leg and left behind leg arises.

(7) Right behind leg and left front leg fall to ground.

(8) Repeats back the process from step (3)

Figure 3.20: The movement style for robot to turn left (cont).

3.3 SOFTWARE IMPLEMENTATION.

For software part, the algorithm program based on high level language is written down using MicroCode Studio. PICkit2 functions as a medium to program the high level language program into microcontroller PIC16F877A.

3.3.1 ALGORITHM AND PROGRAMMING.

The controller will control the entire servo that attach to the leg. The PWM is sent through parallel output port to the signal wire of the servo. Sensor that detects obstacle will send in signal to the controller through input port. The input that occurs at the microcontroller will run the program that makes the robot turn left. It functions to avoid the obstacle.

An algorithm was developed to make the microcontroller to perform the movement of the leg by controlling the servos. Therefore, the algorithm was established and represent by a flow chart below in Figure 3.21.

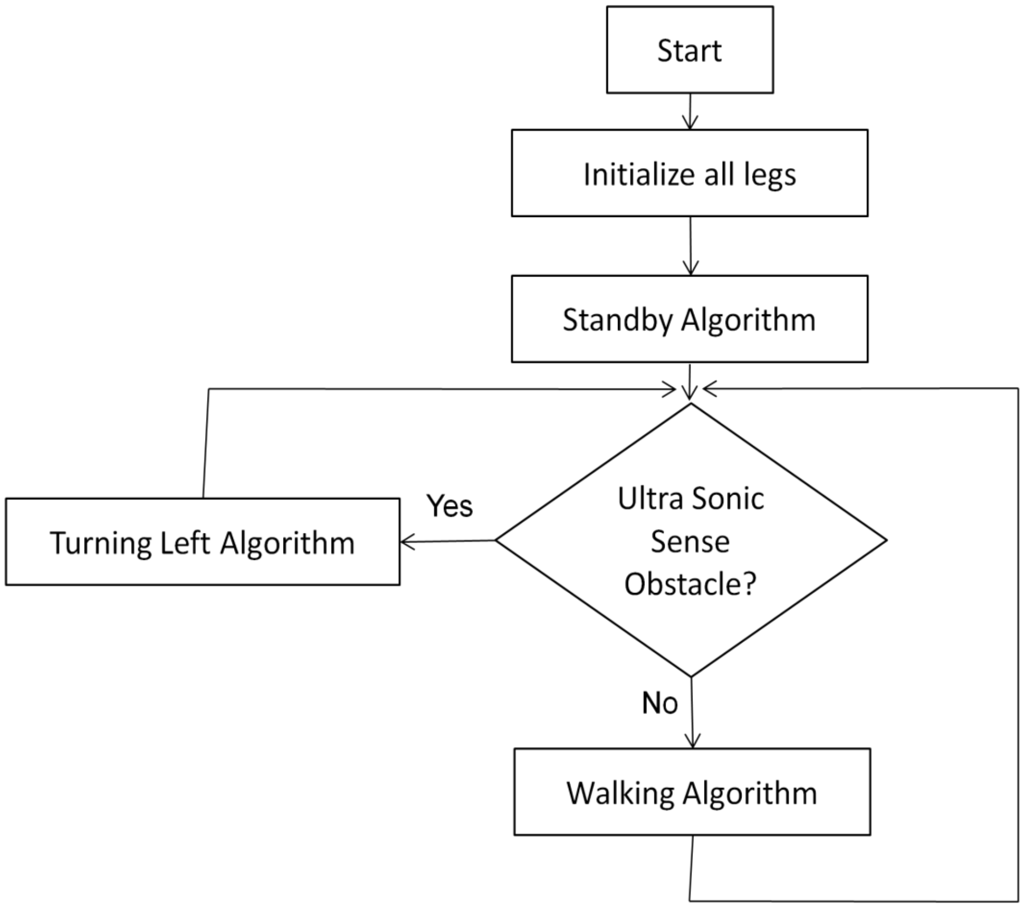


Figure 3.21: General flow chart of program.

3.3.2 PROCESSING EXPLANATION OF MAIN PROGRAM.

There are several main part program was develop for the controller to control the leg of the robot. There are defining oscillation, initialization of all port, defining ADC inputs, PWM generation, standby of all leg, walking straight with sensor scanning, and turn left. Discussion made for the initializing, PWM generation, ADC, and sensor part.

3.3.2.1 INITIALIZE.

Before starting off with the program, variables need to be defined and addressed, it's to make the programming easier to track and remember during writing the program.

```
DEFINE osc 8
DEFINE ADC_BITS 8
DEFINE ADC_CLOCK 3
DEFINE ADC_SAMPLEUS 50

adcon0=%11000101
adcon1=%00001001

counter VAR BYTE
su0 VAR WORD

trisc   = 0
trisd.2 = 1
```

Figure 3.22: Name and variable.

Figure 3.22 shows the initializing part of the program. Based on the program, the value of oscillator defined as 8MHz clock. Then the ADC function is used as an input from the ultrasonic sensor. The defining ADC_BITS is to set number of bits in result, ADC_CLOCK is to set clock source, and ADC_SAMPLEUS for setting sampling time in microsecond. Then the adcon0 and adcon1 are register value and been set according to the project requirement on which port need to be used for ADC and choosing either to receive it digitally or in analog. TrisC defined all 0 means it function the whole portC as output because all the PWM signal that generated and sent to the servo using this PortC. TrisD.2 defined 1 means it functions as an input where it is connected to a switch that makes the robot starts to walk from the standby mode.

3.3.2.2 PWM GENERATION.

There is plenty of ways to generate a PWM pulsing using the PIC Basic Pro. PWM coding method, HIGH LOW coding method, and PULSOUT coding method are one of the common methods to generate the PWM. But for the servo motor, it is best to use PULSOUT method. There is a few things need to be considered before start programming using this method.

PULSOUT Pin, Period

This is a basic PULSOUT program. This command works upon toggling the previous pulse on the pin and toggling it back to generate a perfect PWM. Means, it might not be that stable if the previous pulse is 1 in the pin. So, before this command, a LOW Pulsing is defined for all PortC. The pin could simply define by which port that needs to be generated. The resolution of PULSOUT is dependent upon the oscillator frequency. The high value pulse is successfully generated and for the low value part, command PAUSE was used. The value entered at the pause, is counted in millisecond. Then the period term, take an example:

LOW PORTC

PULSOUT PORTC.0, 300

PAUSE 19

These calculations showed below are based according to the PICBasic Pro system. Since our oscillator frequency is 8MHz.

$$\frac{8MHz}{40} = 0.2MHz \text{--} (1)$$

$$t = \frac{1}{f} = \frac{1}{0.2MHz} = 5\mu s \text{--} (2)$$

$$Peak\ Time\ Cycle = 5\mu s \times 300 = 1.5\ ms \text{--} (3)$$

So this pulse alone will look like this Figure 3.23. Using this pulse the servo will move to 90° straight from any location of the servo, refer Chapter 2 (2.2.1). To make the servo to move to another location, just change the value of 300 to acquire another degree of the servo.

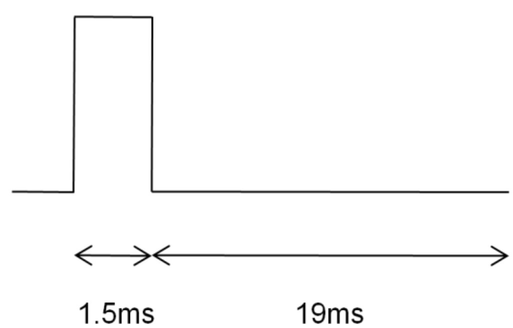


Figure 3.23: Example pulse generation

3.3.2.3 ANALOG TO DIGITAL CONVERTER

In this project, the analog digital converter (ADC) applied as the input at controller that source output from the ultrasonic sensor. The output that produced from ultrasonic sensor is only around 2 Volt. So, it couldn't be used as a normal input since the input needs minimum of 4.5 Volt. That's why the ADC applied in this project. Initializing the ADC was discussed at subchapter 3.3.2.1.Figure 3.24 shows the ADC program that been used.

```
walking:

ADCIN 0,su0

'walking algorithm
IF su0>102 THEN turn

FOR counter = 0 TO 20
LOW portc
PULSOUT portc.1, 120
PULSOUT portc.5, 120
PULSOUT portc.3, 120
PULSOUT portc.7, 180
PAUSE 19
NEXT
```

Figure 3.24: ADC application in the main program.

The ADCIN in the figure 3.34 program shows that the pin 0 at portA is defined for a length of a byte, since it was initialized su0 as byte at initializing program. Then the su0 must be bigger compared to 102 represents as a decimal number. The number 102 decimal is calculated by:

$Received\ Voltage = 2V$ -- (4)

$Refference\ voltage = 5V$ --- (5)

$\frac{2V}{5V} \times 255 = 102$ -- (6)

3.3.2.4 SENSOR DETECTION.

The detection module is so closely related with ADC module in previous. Robot that walks straight has plenty if possibility to face obstacle therefore the need of ultrasonic sensor detector was discussed. Ultrasonic will monitor in front of the robot. In the program every time the robot make the movement to walk straight it will scan either there is obstacle in the in front or not, if the obstacle occurs the robot will turn left. After the turn left algorithm program executed, it will continue with walking straight routine.

The movement of the walking straight and turning left that implement to the program is based on the movement design in the robot movement design that was discussing before. In Figure 3.25 is the flow chart of the program of the robot walking straight. The coding program implementation is at the Figure 3.24. The input that received is connected to the ADC.

IF su0 > 102 **THEN** turn

This is the command that makes the robot able to turn left when it meets an obstacle in front. The ultrasonic detects, then produces 2V as an output from it then enters the controller using the ADC feature. The ADC command been written in such way if any 2V in that particular pin enters, then this command will be executed.

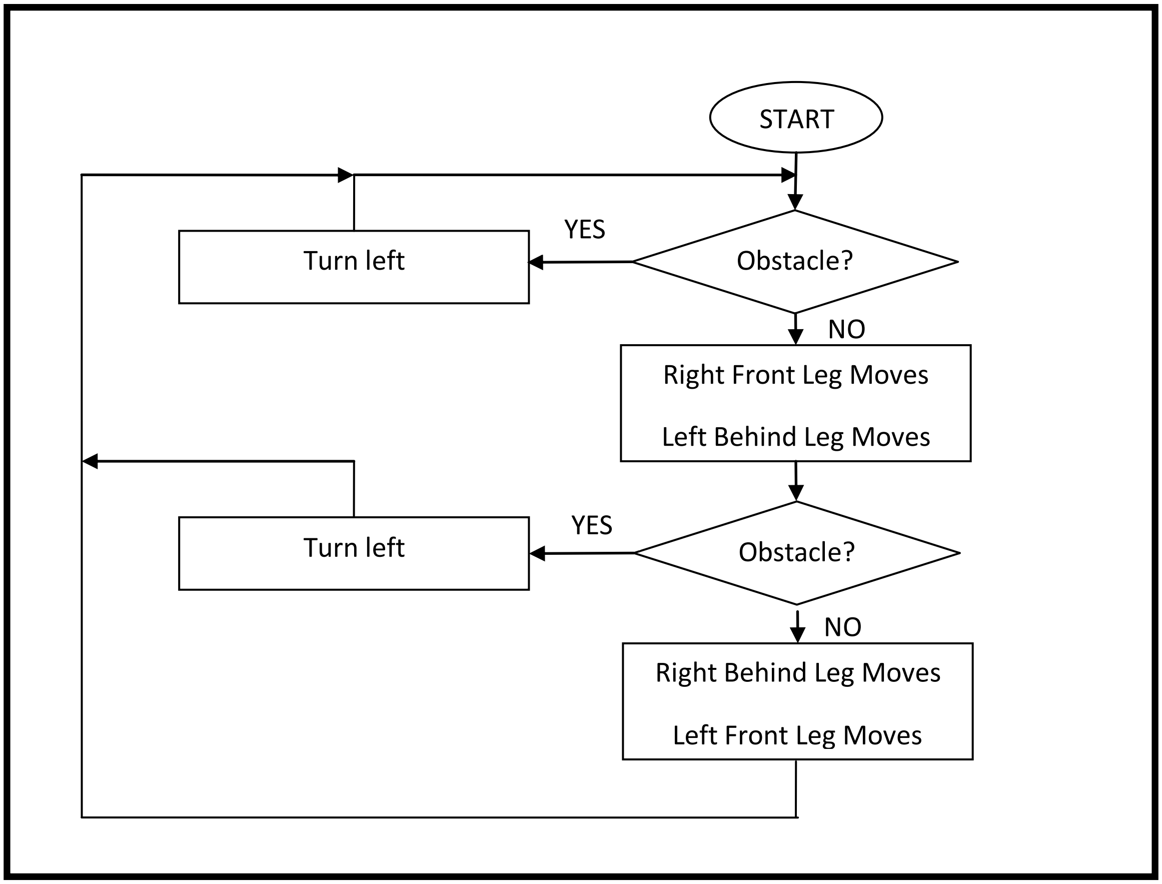


Figure 3.25: Walking straight and the sensor detection scanner flow chart.

CHAPTER 4

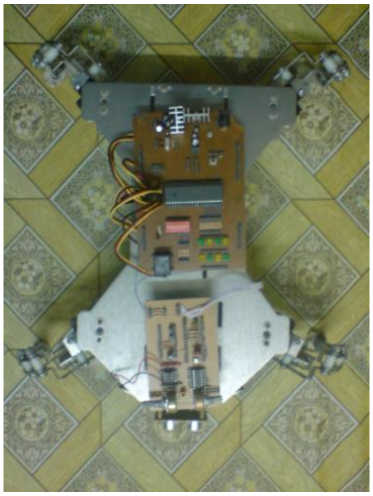
RESULT AND ANALYSIS.

4.1 INTRODUCTION.

The result after completion of this project is successful as accordance to the objective and the scope of the project. This complete quadruped robotic system fully controlled by the PIC16F877A microcontroller. From the sensing module until the movement of every leg of the robot that is attached to the controller so that the system could be successful. The quadruped robot is able to walk straight, make turn and detect an obstacle ahead.

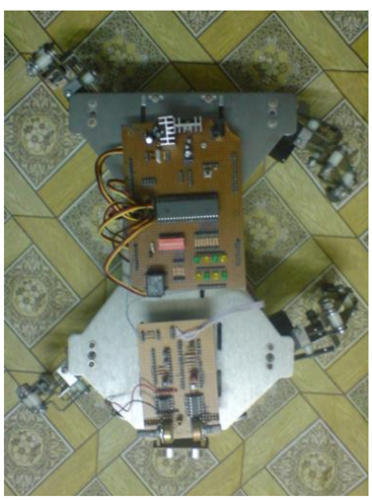
4.2 TOTAL RESULT ROBOT ALGORITHM.

After developing the entire module and ensuring the project is ready, the robot finally tested on the floor. Placing it at the floor, and the power is turned ON to see the movement of the robot after program were install into the microcontroller based on the movement that had been design. The picture in Figure 4.0 is from the observance of the movement robot to walk straight ahead. Figure 4.1 and Figure 4.2 shows the turning left after occurrence of an obstacle ahead of it.



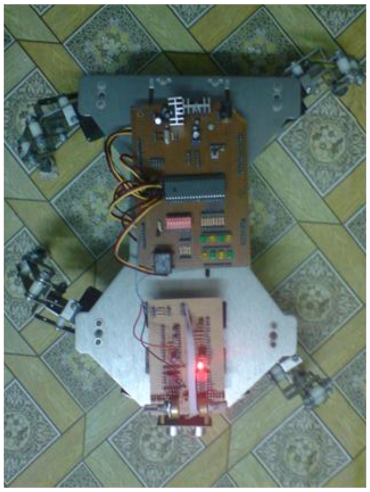
(a) Straightening all the legs into initial condition.
(b) The robot is in the standby mode so that it is ready to walk.

(c) All the leg moves according to the program.
(d) The robot manages to go one-step ahead.

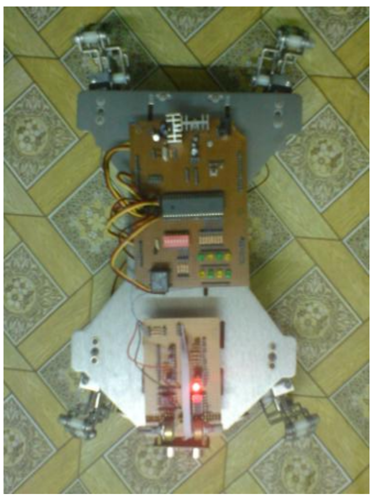


(e) The robot in the second step goes one more step ahead.

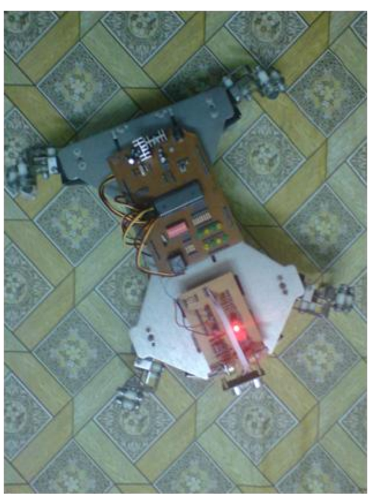
Figure 4.0: Result robot walk straight.



(a) The robot senses obstacle ahead.
(b) The LED blinks indicates the existence of obstacle.



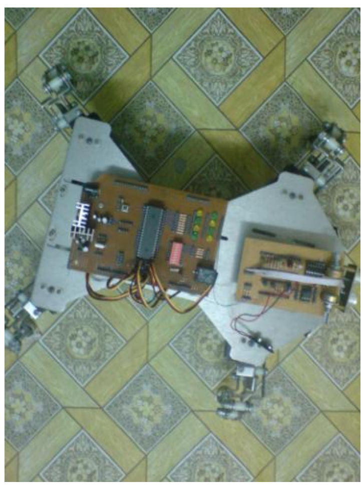
(c) After sensing the obstacle, the module changes to the turning module as the programming.
(d) All the four legs turns to the



(e) The second step of left movement executes.
(f) The robot start moves left slowly.

Figure 4.1: Result of turning left after detection of the obstacle.

(g) The robot continues the second step to turn left since the obstacle still exists.



(h) After the final step of left movement, the LED stopped blinking.

(i) This condition will pursue back to part (c) of walking straight

Figure 4.2: Result of turning left after detection of the obstacle (cont).

4.3 MATHEMATICAL EXSPRESSION OF THE LEG IN MATRIX FORM.

To express the movement of the leg of the robot in mathematical expression, Denavit- Hartenberg representation will be used to derive the equation of the motion of the one of the leg of robots. The robot leg can be represent in Figure 4.3 below.

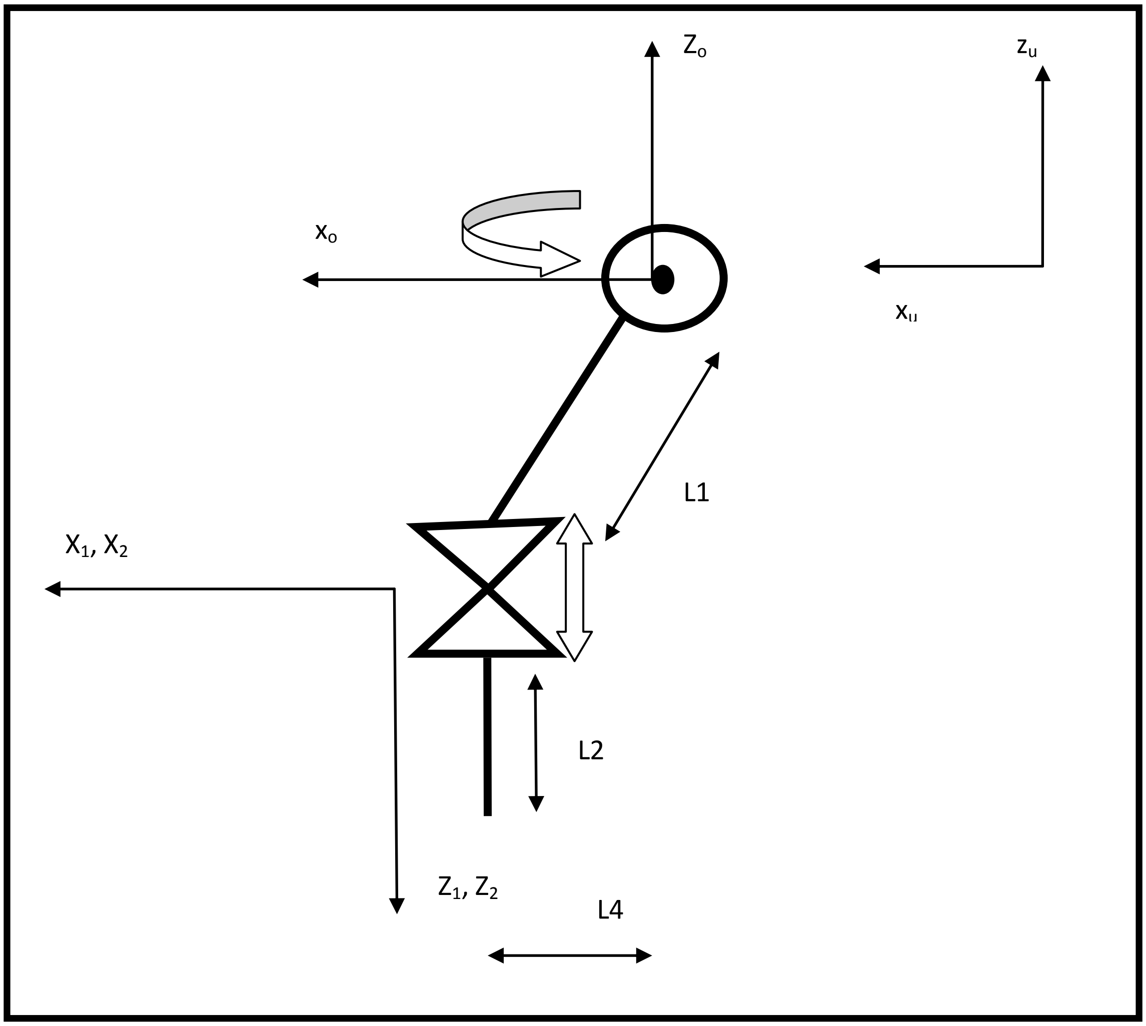


Figure 4.3: Leg represent in robotic analysis.

Table 4.0: D-H Parameter.

#	θ	*d*	*a*	α
1	0°	L1	L4	-180°
2	0°	L2	0	0°

$$A_{n+1} = \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1}C\alpha_{n+1} & S\theta_{n+1}S\alpha_{n+1} & a_{n+1}C\theta_{n+1} \\ S\theta_{n+1} & C\theta_{n+1}C\alpha_{n+1} & -C\theta_{n+1}S\alpha_{n+1} & a_{n+1}S\theta_{n+1} \\ 0 & S\alpha_{n+1} & C\alpha_3 & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{------------------------} \; (1)$$

$$A_1 = \begin{bmatrix} \cos(0) & -\sin(0)\cos(-180) & \sin(0)\sin(-180) & L4\cos(0) \\ \sin(0) & \cos(0)\cos(-180) & -\cos(0)\sin(-180) & L4\sin(0) \\ 0 & \sin(-180) & \cos(-180) & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{-----------} \; (2)$$

$$\therefore A_1 = \begin{bmatrix} 1 & 0 & 0 & L4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{--} \; (3)$$

$$A_2 = \begin{bmatrix} \cos(0) & -\sin(0)\cos(0) & \sin(0)\sin(0) & (0)\cos(0) \\ \sin(0) & \cos(0)\cos(0) & -\cos(0)\sin(0) & (0)\sin(0) \\ 0 & \sin(0) & \cos(0) & L2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{----------------------} \; (4)$$

$$\therefore A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{--} \; (5)$$

$^{U}T_{H} = A_u\, A_1 A_2$

$$^{U}T_{H} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & L4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{--------------------------- (6)}$$

$$\therefore\ ^{U}T_{H} = \begin{bmatrix} 1 & 0 & 0 & L4 + x \\ 0 & -1 & 0 & y \\ 0 & 0 & -1 & L1 + L2 + z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{-- (7)}$$

Where x, y, z are distance from universe axis.

CHAPTER 5

CONCLUSION AND RECOMMENDATION.

5.1 CONCLUSION.

Almost everything in our daily life based on robotics design. Robotics is getting advance and higher performance by day. It is used in various type of field and could be found in many industries that required robotic to do complicated and dangerous kind of job. For an example at a car production line, there are robots that weld to design and complete a car. Thus, there is another scale of robotic development that is more towards inspiration from biological insects and animals. Therefore, the project does not develop a robot that helps and do a job for us but more towards reverse engineering from the insects and animals. Understanding this process, then only this project was started to be developed.

The goal of this project is to design an algorithm and to control the movement of 4- legged robot by using PIC 16F877A microcontroller so it can walk, make turn based on animals' movement and avoid obstacle. The movement of the leg can be control by varying the PWM signal from microcontroller to the servo motor that attach to the robot leg.

In conclusion, the robot is able to walked, make turns and avoid obstacle as designed, and there is still flaw need to be improvised. The objective of the project is successfully fulfilled.

5.2 PROBLEMS AND SOLUTION.

During the process of completing this project, there are many problems occur. The problems are:

a) The shoulder is not capable to rotate 0° and 180° position since it will hit the robot body, therefore in the program the servo were only set to rotate 30° and 150° position so the leg won't hit the body.

b) Power consumption, servomotor requires high current due to the load its carry. If it is only about controlling one servomotor, 500mAh should be fine and normal AA batteries can produce enough current for long time. But there is several servos that need to be controlled, total of 8 servos does consist of high current. For rechargeable lead acid batter which is have high current were used to solve the problem but didn't last long since the disadvantage of this batteries it was heavy compare to normal AA batteries. The heavy of this battery alone made the robot is not that capable to walk smoothly and does produce a severe sound. Finally a direct power supply is been used. Nevertheless, the supply still could not drive enough current for the 8 servos. Then another supply is been added to the circuit and finally the robot crawls smoothly. A separation of 4 servo for one supply and another 4 servo for another supply.

c) The voltage that produced by the ultrasonic sensor is not enough for the PIC to detect as an input. This case happens after the double power supply is been developed for the robot. Once all the grounding of two supplies and a sensor is attached together, the output that previously producing 8V all of a sudden dropped until 2V surprisingly. Finally, the solution was to use the ADC feature inside the PIC controller. It was set to trigger the turning algorithm once the ADC detects 2V and above. This solution was the best among all that been tried previously to troubleshoot the problem.

d) The program of the robot is a bit tricky to be developed. This is stated since there is no such existence of 18.5ms for a pause command, but our servo needs a full 20ms of duty cycle in order to move to that particular angle. By this, the program still sticks to the 19ms but manipulation only at the high pulse of it. A bit different program and setting been done but it still works.

5.3 RECOMMENDATION.

The performance of the system is slightly sluggish and unstable. For the future works, there are some recommendations that been listed based on the problems in order to improve the performance.

a) PID control system need to be implemented to get a better performance and smoother in walking movement.

b) Use a robot mapping for the robot so that after it could plan the journey itself before even start moving.

c) Equip many other expensive sensors to make the robot move according the change of the surrounding and can collect precise data.

d) Change the style from not only able to move and avoid the obstacle, but it could even do some useful application that could help and improvise our daily life style.

5.4 COSTING AND COMMERCIALIZATION

Any project that developed, commercializing is important so that the project that developed does have the purpose of contributing to technology and our daily life. The costing of the whole project listed down in the Table 5.0. However, the values are not 100% accurate and just a rough figure since the project does have a lot of addition of free items and subsidy by the faculty laboratory.

Table 5.0: Costing of the project

Items	**Price (RM)**
Quadruped Body Robot	1600.00
Sensor development	40.00
Main board	80.00
Additional apparatus	50.00
Total	**RM 1790.00**

This project based on research and developed of biological inspired robot. Means this project is towards developing a robot that could act and move like an insect of animal. Commercializing of this project itself is not suitable since the implementation to our daily life is null. However, this function and the research that been developed throughout this project could be an extensive device to another device so that it could be widely used and commercialized. Example, this robot could crawl safely at any surface. If an extension of a sucker used at every leg of the robot, with the proper balancing and movement, we could attach a glass cleaner in front of the robot. So, that each step it crawls, it could wipe and clean. The horizontally fixed glass at some shopping complexes use sometimes gets dirty and a person to clean it might be dangerous. Therefore, this robot could act as the cleaner. In an advance level, it might be able to climb up the window that stands vertically and wipe it shining clean.

REFERENCE.

[1] About Servo: The specifications, 2008

Available at: http://www.servocity.com/html/hs-322hd_standard_deluxe.html

[2] MicroEngineering Labs, Inc.(2004) PicBasic Pro Complier.

[3] Saeed B. Niku, Prentice Hall (2004). Introduction to Robotics Analysis, System, Application.

[4] Alexander Sadiku, McGraw Hill (2007), Fundamental of Electric Circuits third edition.

[5] Yvonne So, Ryerson Polytechnic University (1998), Legged Robot.

[6] Abdul Hadi Mohamad, Thesis (2007), Multi Legged Robot, Inspire By Nature

[7] John Iovine, McGraw-Hill (2000), PIC Microcontroller Project Book

APPENDIX A

SOFTWARE DEVELOPMENT.

```
'*****************************************************************

'*  Name    : quadruped walking robot.BAS                        *
'*  Author  : [Mohamed Faizan Bin Basheer Ahmad                  *
'*  Notice  : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*          : All Rights Reserved                                *
'*  Date    : 10/15/2008                                         *
'*  Version : 1.0                                                *
'*  Notes   :                                                    *
'*          :                                                    *
'****************************************************************

define osc 8
Define ADC_BITS 8
Define ADC_CLOCK 3
Define ADC_SAMPLEUS 50

adcon0=%11000101
adcon1=%00001001

counter var byte
su0 var word


trisc   = 0
trisd.2 = 1


main :
adcin 0,su0

if portd.2 = 1 then walking

'early standby

'raise all

for counter = 0 to 20
low portc
pulsout portc.1, 400
pulsout portc.3, 120
pulsout portc.5, 400
pulsout portc.7, 180
pause 19
next

'sit position

for counter = 0 to 20
low portc
pulsout portc.3, 400
pulsout portc.5, 120
pause 19
next
```

```
'adjust front left
for counter = 0 to 20
low portc
pulsout portc.7, 450
pulsout portc.6, 240
pause 19
next

'adjust front right
for counter = 0 to 20
low portc
pulsout portc.1, 120
pulsout portc.0, 380
pause 19
next

'raise behind
for counter = 0 to 20
low portc
pulsout portc.3, 120
pulsout portc.5, 400
pause 19
next

'adjust behind right
for counter = 0 to 20
low portc
pulsout portc.3, 400
pulsout portc.2, 250
pause 19
next

'adjust behind left
for counter = 0 to 20
low portc
pulsout portc.5, 120
pulsout portc.4, 340
pause 19
next

'raise all
for counter = 0 to 20
low portc
pulsout portc.3, 120
pulsout portc.5, 400
pulsout portc.1, 400
pulsout portc.7, 180
pause 19
next

pause 500

'sit still
```

```
for counter = 0 to 20
low portc
pulsout portc.1, 120
pulsout portc.3, 400
pulsout portc.5, 120
pulsout portc.7, 450
pause 19
next



pending:
if portd.2 = 1 then walking
goto pending

end



walking:

adcin 0,su0

'walking algorithm
if su0>102 then turn

for counter = 0 to 20
low portc
pulsout portc.1, 120
pulsout portc.5, 120
pulsout portc.3, 120
pulsout portc.7, 180
pause 19
next

if su0>102 then turn

for counter = 0 to 20
low portc
pulsout portc.0, 380
pulsout portc.4, 240
pulsout portc.2, 250
pulsout portc.6, 340
pause 19
next

if su0>102 then turn

for counter = 0 to 20
low portc
pulsout portc.1, 400
pulsout portc.5, 400
pulsout portc.3, 400
```

```
pulsout portc.7, 450
pause 19
next

if su0>102 then turn

for counter = 0 to 20
low portc
pulsout portc.0, 280
pulsout portc.4, 340
pulsout portc.2, 350
pulsout portc.6, 240
pause 19
next

goto pending



turn:



for counter = 0 to 20
low portc
pulsout portc.1, 120
pulsout portc.5, 120
pulsout portc.3, 120
pulsout portc.7, 180
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 450
pulsout portc.4, 450
pulsout portc.2, 150
pulsout portc.6, 150
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.1, 400
pulsout portc.5, 400
pulsout portc.3, 400
pulsout portc.7, 450
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 400
pulsout portc.4, 400
```

```
pulsout portc.2, 200
pulsout portc.6, 200
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.1, 120
pulsout portc.5, 120
pulsout portc.3, 120
pulsout portc.7, 180
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 450
pulsout portc.4, 450
pulsout portc.2, 150
pulsout portc.6, 150
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.1, 400
pulsout portc.5, 400
pulsout portc.3, 400
pulsout portc.7, 450
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 400
pulsout portc.4, 400
pulsout portc.2, 200
pulsout portc.6, 200
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.1, 120
pulsout portc.5, 120
pulsout portc.3, 120
pulsout portc.7, 180
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 450
pulsout portc.4, 450
```

```
pulsout portc.2, 150
pulsout portc.6, 150
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.1, 400
pulsout portc.5, 400
pulsout portc.3, 400
pulsout portc.7, 450
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 400
pulsout portc.4, 400
pulsout portc.2, 200
pulsout portc.6, 200
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.1, 120
pulsout portc.5, 120
pulsout portc.3, 120
pulsout portc.7, 180
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 450
pulsout portc.4, 450
pulsout portc.2, 150
pulsout portc.6, 150
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.1, 400
pulsout portc.5, 400
pulsout portc.3, 400
pulsout portc.7, 450
pause 19
next

for counter = 0 to 20
low portc
pulsout portc.0, 400
pulsout portc.4, 400
```

```
pulsout portc.2, 200
pulsout portc.6, 200
pause 19
next


goto walking
```

APPENDIX B

MICROCHIP PIC16F877A DATA SHEET.

LM7805 DATA SHEET.

HITEC HD-322HD DATA SHEET

MOTOROLA MC68HC11 DATA SHEET.

MICROCHIP

PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

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 other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

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
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^{2}C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

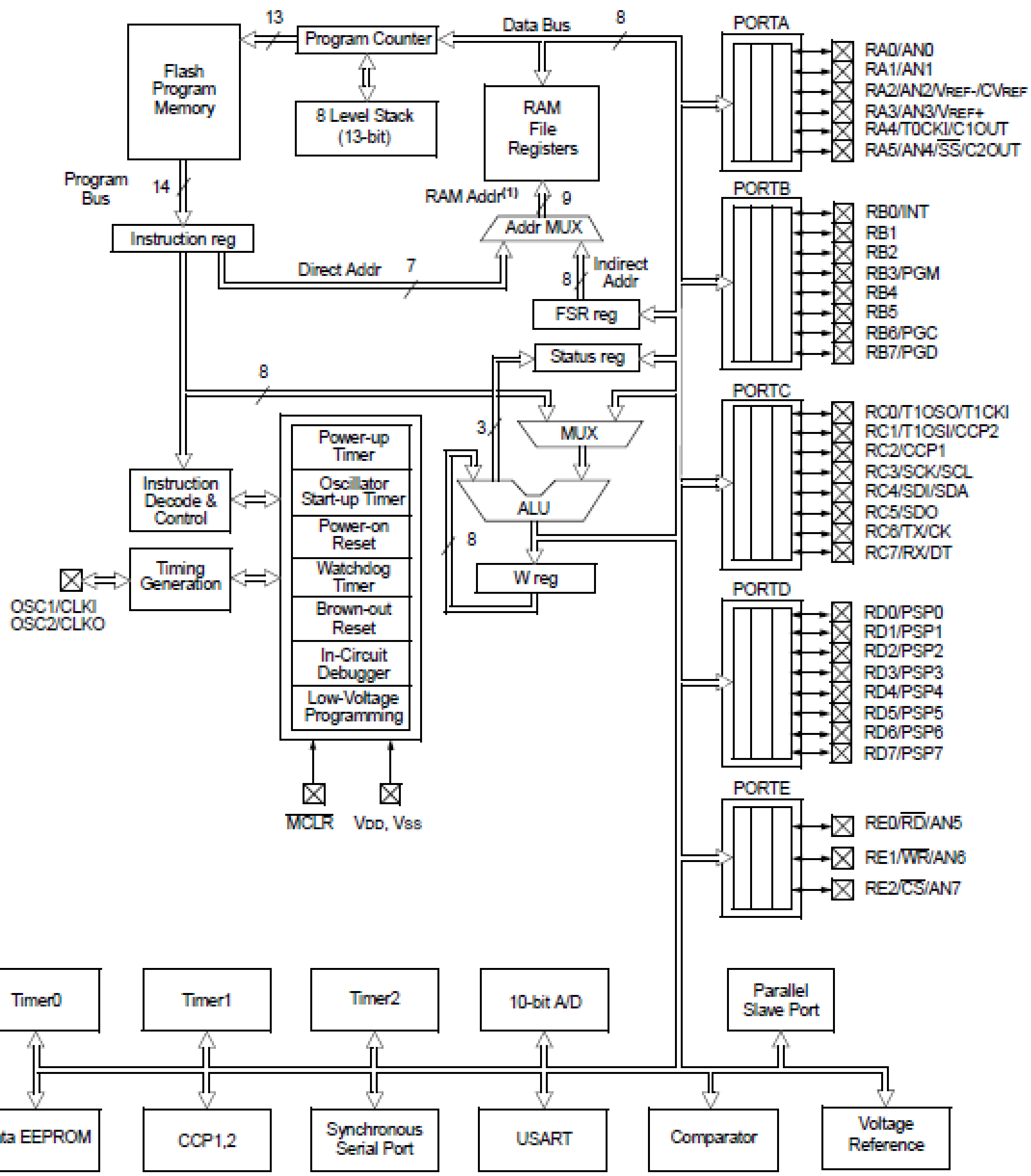
Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

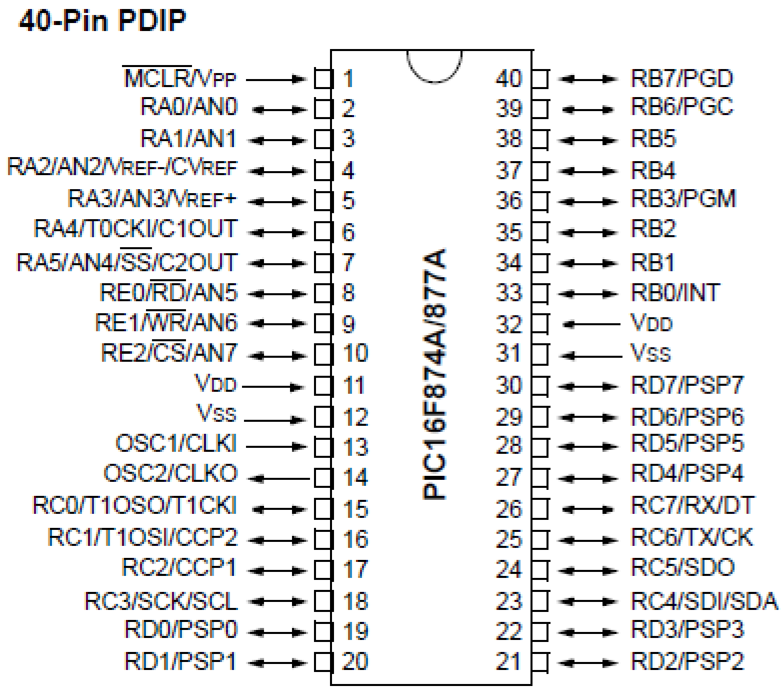
- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^{2}C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2



Device	Program Flash	Data Memory	Data EEPROM
PIC16F874A	4K words	192 Bytes	128 Bytes
PIC16F877A	8K words	368 Bytes	256 Bytes

Block Diagram of PIC16F877A


40-Pin PDIP
PIC16F874A/877A
MCLR/VPP 1
RA0/AN0 2
RA1/AN1 3
RA2/AN2/VREF-/CVREF 4
RA3/AN3/VREF+ 5
RA4/T0CKI/C1OUT 6
RA5/AN4/SS/C2OUT 7
RE0/RD/AN5 8
RE1/WR/AN6 9
RE2/CS/AN7 10
VDD 11
VSS 12
OSC1/CLKI 13
OSC2/CLKO 14
RC0/T1OSO/T1CKI 15
RC1/T1OSI/CCP2 16
RC2/CCP1 17
RC3/SCK/SCL 18
RD0/PSP0 19
RD1/PSP1 20
40 RB7/PGD
39 RB6/PGC
38 RB5
37 RB4
36 RB3/PGM
35 RB2
34 RB1
33 RB0/INT
32 VDD
31 VSS
30 RD7/PSP7
29 RD6/PSP6
28 RD5/PSP5
27 RD4/PSP4
26 RC7/RX/DT
25 RC6/TX/CK
24 RC5/SDO
23 RC4/SDI/SDA
22 RD3/PSP3
21 RD2/PSP2

LM7805 DATA SHEET.

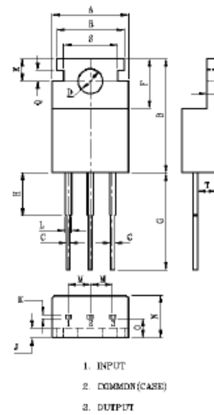
THREE TERMINAL POSITIVE VOLTAGE REGULATORS
5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V, 20V, 24V.

FEATURES

- Suitable for C-MOS, TTL, the Other Digital IC's Power Supply.
- Internal Thermal Overload Protection.
- Internal Short Circuit Current Limiting.
- Output Current in Excess of 1A.
- Satisfies IEC-65 Specification. (International Electronical Commission).

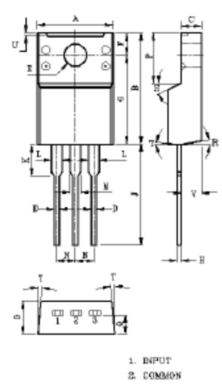
MAXIMUM RATINGS (Ta=25℃)

CHARACTERISTIC		SYMBOL	RATING	UNIT
Input Voltage	KIA7805AP/API~ KIA7815AP/API	V_{IN}	35	V
	KIA7818AP/API~ KIA7824AP/API		40	
Power Dissipation (Tc=25℃)		P_D	20.8	W
Power Dissipation (Without Heatsink)	KIA7805API~ KIA7824API	P_D	2.0	W
Operating Junction Temperature		T_j	-30~150	℃
Storage Temperature		T_{stg}	-55~150	℃



DIM	MILLIMETERS
A	10.30 MAX
B	15.30 MAX
C	4.80
D	φ3.60± 0.30
E	3.00
F	6.70 MAX
G	13.60± 0.50
H	5.60 MAX
J	1.37 MAX
K	0.50
L	1.60 MAX
M	2.54
N	4.70 MAX
O	2.40
P	1.30 MAX
Q	1.50
R	9.50± 0.20
S	8.00± 0.20
T	2.90 MAX

TO-220AB



DIM	MILLIMETERS
A	10.30 MAX
B	15.30 MAX
C	2.70±0.30
D	0.85 MAX
E	φ3.20±0.20
F	3.60±0.30
G	12.50 MAX
H	0.75 MAX
J	13.60±0.50
K	3.90 MAX
L	1.20
M	1.50
N	2.54
O	4.50±0.20
P	6.80
Q	2.60±0.20
R	10°
S	25°
T	5°
U	0.5
V	2.60±0.15

TO-220IS

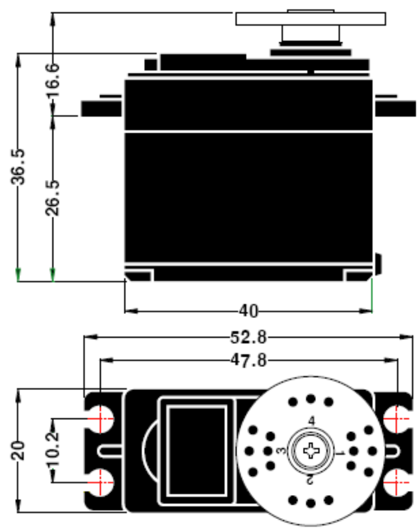
HITEC HD-322HD DATA SHEET

PREPARED BY JUN HEE, LEE
UPDATE:APR 01, 2002

ANNOUNCED SPECIFICATION OF HS-322HD STANDARD DELUXE SERVO

1.TECHNICAL VALUE

CONTROL SYSTEM	:+PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	:4.8V TO 6.0V	
TEST VOLTAGE	:AT 4.8V	AT 6.0V
OPERATING SPEED	:0.19sec/60? AT NO LOAD	0.15sec/60? AT NO LOAD
STALL TORQUE	:3kg.cm(41.66oz.In)	3.7kg.cm(51.38oz.In)
IDLE CURRENT	:7.4mA AT STOPPED	7.7mA AT STOPPED
RUNNING CURRENT	:160mA/60? AT NO LOAD	180mA/60? AT NO LOAD
STALL CURRENT	:700mA	800mA
DEAD BAND WIDTH	:5usec	5usec
OPERATING TRAVEL	:40? /ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	:CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
MOTOR TYPE	:CORED METAL BRUSH	
POTENTIOMETER TYPE	:4 SLIDER/DIRECT DRIVE	
AMPLIFIER TYPE	:ANALOG CONTROLLER & TRANSISTOR DRIVER	
DIMENSIONS	:40x20x36.5mm(1.57x0.78x1.43In)	
WEIGHT	:43g(1.51oz)	
BALL BEARING	:TOP/RESIN BUSHING	
GEAR MATERIAL	:2 HEAVY DUTY RESIN	
HORN GEAR SPLINE	:24 SEGMENTS/?5.76	
SPLINED HORNS	:RESULAR/R-C,R-D,R-I,R-O,R-X,SUPER/R-XA	
CONNECTOR WIRE LENGTH	:300mm(11.81In)	
CONNECTOR WIRE STRAND COUNTER	:40EA	
CONNECTOR WIRE GAUGE	:25AWG	



2.FEATURES

LONG LIFE POTENTIOMETER, TOP RESIN BUSHING, 2 HEAVY DUTY RESIN GEARS

3.APPLICATIONS

AIRCRAFT 20-40 SIZE,STEERING AND THROTTLE SERVO FOR CARS, TRUCK AND BOATS

4.ACCESSORY & OPTION

CASE SET/	GEAR SET/	HORN SET/
HS322T:1EA	HS322G1:1EA	R-C:1EA
HS322M:1EA	HS325G2:1EA	R-D:1EA
HS322L:1EA	HS325G3:1EA	R-I:1EA
PH/T-2 2x30 NI:4EA	HS322G4:1EA	R-O:1EA
	HS300RB:1EA	R-X:1EA
		R-XA:1EA
		WH/W 2.1x15 NI:4EA
		BST 3x5.5:4EA
		NBR 9x6.5x6:4EA