An autonomous robot for waiter service

in restaurants

Mohd. Abdullah Omair ………………………………………… 11121094

Al Shahriar Hossain Rakib ……………………………….…... 11121042

Md. Amir Khan ……………………………………………… 11121093

Rubaiyat Talat Mahmud …………………………………...... 11121106

Department of Electrical and Electronic Engineering

BRAC University, Dhaka, Bangladesh

SPRING 2015

Declaration

We hereby declare that the Thesis Title "An autonomous robot for waiter service in restaurants" is submitted to the Department of Electrical and Electronic Engineering of BRAC University in partial fulfillment of the Bachelor of Science in Electrical and Electronic Engineering. This is our original work and was not submitted elsewhere for the award of any other degree or any other publication.

Date:

____

Dr. Md. Khalilur Rahman

Associate Professor

Thesis Supervisor

____

Mohd. Abdullah Omair

____

Al – Shahriar Hossain Rakib

____

Md. Amir Khan

____

Rubaiyat Talat Mahmud

Acknowledgement

We would like to take this opportunity to express our deepest gratitude and appreciation to BRAC University for their support with which we were able to complete this Thesis project. The skills and ideas we have been able to learn from our Thesis work have benefited us immensely and will continue to do so throughout our future endeavors.

Abstract

The Waiter-Bot is an exceptional autonomous robot which has the ability to follow a designated path like a pathfinder with the help of IR sensor arrays and reach its intended destination. It is an arduino based robotic design implemented to seek out and detect its required destination and perform its deliberate tasks with precision and accuracy. The Waiter-Bot consists of simple mechanical design which has simple mechanism with which performing the necessary tasks becomes easier. Due to its design parameters, it also requires less power and draws less current which allows us to work with the robot safely. Moreover, it also adds to the safety to its environment and the people around it. Its light weight build up allows us to assemble the parts without having the difficulty to carry it anywhere. Once assembled, the entire robot still remains light in its weight and can easily be moved around if necessary. Based on its complete set-up and overall parameters, the waiter-bot stands out on its own to become an important aspect to the solution of this thesis project.

Table of Contents:

Chapter 1: Introduction:

1.1Literature Review:

In Bangladesh, many restaurants which have good quality and standard to its credentials sometimes lose its charismatic reputation and business status because of certain facts overlooked by their entrepreneurs. Facts such as time maintained quality service, human errors during service i.e. serving an order to the wrong table, lack of systematic approach are critical to the customers and their satisfaction. It is usual in any Bangladeshi restaurant to observe customers leave unsatisfied and furiously disappointed due to poor waiting service. This is very much visible in restaurant peak hours. To be crystal clear, this noticeable scene happens due to lack of employers during the time when endless customers are storming into the restaurant to enjoy a delicious meal. The employers become puzzled and are unable to manage the proper service which is necessary to be provided to the customers. This consumes a lot of time and hence generates intense service lag which results in disappointment. This provoked us to find an automated solution to this particular problem and hence we came up with the idea of the Waiter-Bot. Not just in Bangladesh, this idea was turned into reality in other countries like China and Japan and showed improved customer rating of the restaurant. Not only that, it also enhanced the revenue earned by the particular restaurant and therefore provided a drastic economic boost. Most of the customers visited the restaurant just to see the robot in function performing its tasks. More and more customers seeped in and so increased the restaurants business status in its respective market. Never the less, for any restaurant to function effectively and stand out in its market, a certain organized system is vital for its survival. Moreover, a well-structured restaurant is liable to attract more customers and provide top standard services to satisfy the customers.

1.2 Motivation:

To the problems observed in Bangladesh relating the respective crisis, we started to look for solutions to this problem and eventually came across the answers we were looking for. We have successfully discovered that in Harbin, Heilongjiang province in China, a restaurant named "Robot Restaurant" have robotic staff to wait the tables. [1] These robots are able to travel around via sensors on a designated path and are capable of serving 30 different dishes and has a battery life of four to five hours. [1]

Fig.1 Robots used in restaurants abroad

The implementation of robots in commercial industries and many other firms has proven itself in providing a very promising future of advanced robotics. We can understand its influence in the industrial market just by seeing how much effort and financial support a particular company gives in making robots such as the waiter-bot. The Harbin Haohai Robot Company has invested a total of 5 million yuan in the mass production of these robots and each robot cost about 200,000 to 300,000 yuans. [1] Based on the details of this finding we decided that a similar project can be implemented in Bangladesh in a cost effective manner. Accumulating industrial and marketing facts, we were also able to realize that this particular project will have a phenomenal effect on the country's economic stability as this Thesis project is attempted for the first time in Bangladesh. If the waiter-bot gets an eye-catching recognition all around

Bangladesh, then this project can further be improved and implemented in a much refined and beneficial manner. Thus the inspiration of waiter-bot gave us hope of turning an innovative idea into a faceable reality.

Chapter 2: Simulation:

2.1 System Architecture:

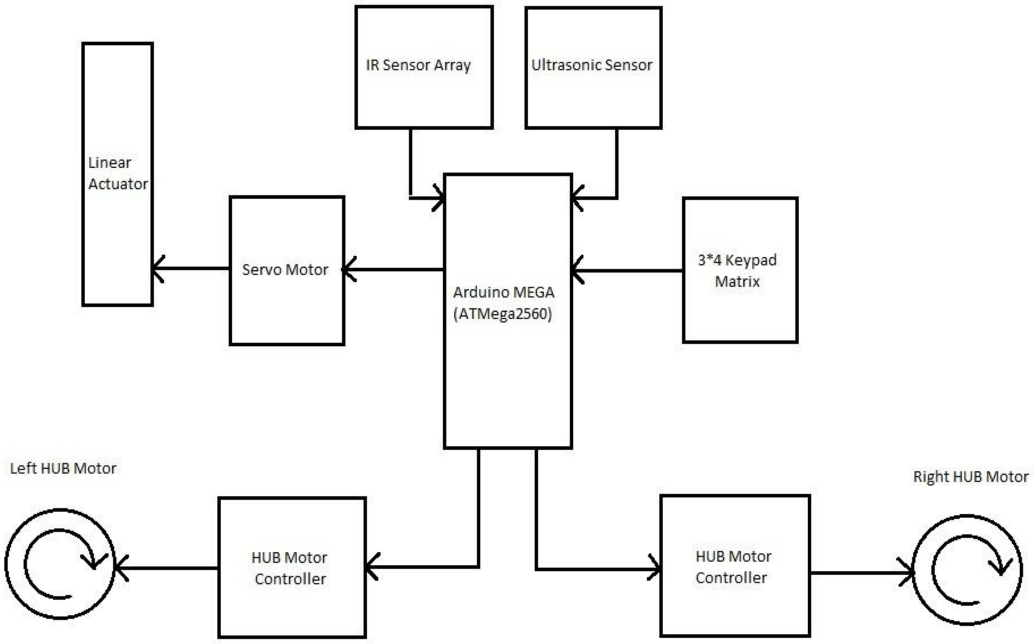
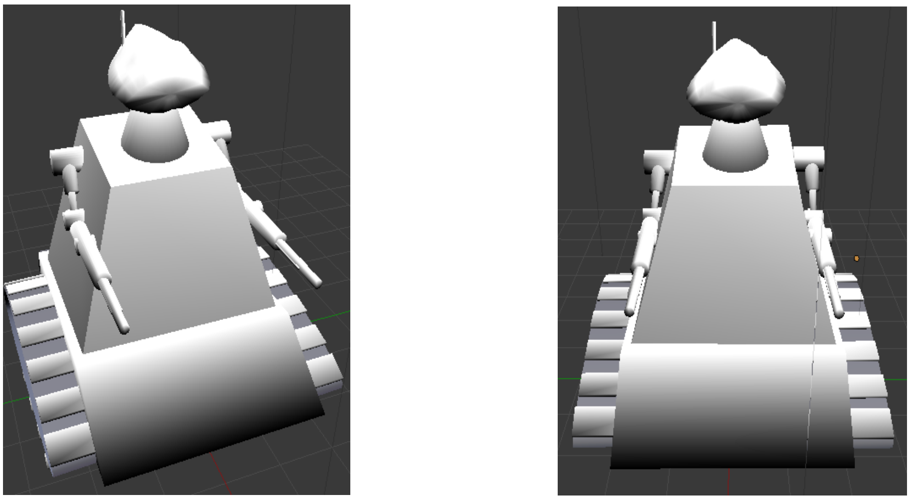


Fig.2 Block Diagram of entire System

The system architecture shown in the figure above at a glance provides us with a clear understanding of how each system was interfaced with each other. As we can visualize, the microcontroller (ATMega2560) receives the keypad input as its destination information. It then signals the HUB motor controller units to run the motor with respect to the IR sensor and the ultrasonic sensor values received respectively. Using these values, the microcontroller detects if any obstacle is in the way and also decides whether the robot has reached its intended destination. Once it reaches its correct position it then signals the servo motor to deviate from its idle position and set-up a connection to the linear actuator hence allowing it to serve the meal. The entire process will further be discussed in the upcoming chapters.

2.2 3D Model:

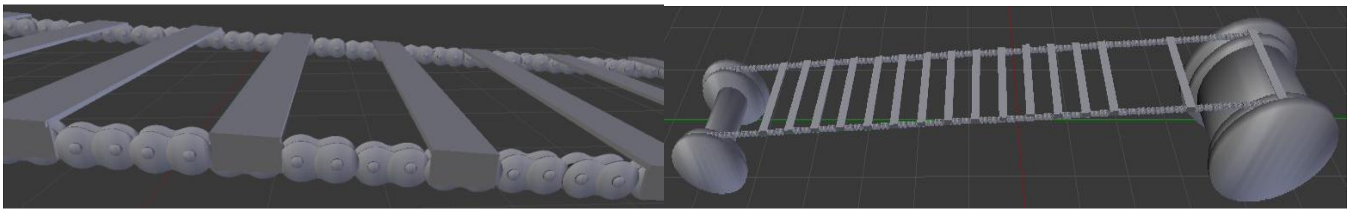
Instead of rushing into an irrational decision of starting the practical work immediately, proper planning and certain steps of approach must be brainstormed beforehand. Hence, we have created a three dimensional software model of the robot. Using the blender software we were able to design the entire robot structure and the mechanical systems involved. It is essential to build a three dimensional model before implementing it as it will give us the essence of the robot and what it might look like. More importantly, it will provide us with an idea whether the robot is faceable to be put into production before making critical errors. In doing so, we were able to evaluate the complexity, possibilities and the delicacy of the project before going in depth into the practical work. The software that we have used is the blender software. Blender is a type of software with which we can make a three dimensional model of any practical implementation based on its criteria. Not only that, we will be able to view the design from any angle and figure out what changes should be if necessary and what problems we might face during implementation. Below are some pictures of the software design.



(a) (b)

Fig. 3 (a) Camera view of the robot   (b) Front view of the robot

Here we can see that our initial design scheme consists of a bicycle chain track system. The reason to use this means of locomotion is that it provides good stability to the robot. This is necessary because if pushed slightly from the sides or from any angle, it provides a virtual stable platform which has a very low centre of mass. As a result tipping over the centre of mass out of balance is very difficult and is an advantage to the waiter-bot according to its environment. Furthermore, a three dimensional model of the locomotion system was also designed and viewed at all possible angles and later have been decided how it will affect the robot accordingly. Below are the snapshots taken from the blender software.



(a) (b)

Fig. 4 (a) Chain track design. (b) Partial design of locomotion system

Using these designs we were able to evaluate what our robot was going to look like. None the less, as the progress goes on we will be able to see clearly how the ultimate design changed according to problems faced during each practical implementation and also we will be able to understand why the changes were necessary.

Chapter 3: Mechanical Implementation:

3.1 Original Design:

3.1.1 Locomotion System:

The rear wheel of the track itself is self-propelling motor which can be driven by connecting it with a micro controller. The motor is specially designed and attached with two spockets screwed to its sides. The spockets are specially cut to maintain required set of teeth to which the bicycle chain will be attached.

Fig. 4 HUB motor with spocket

As it is visible in the picture, two steel bars are first drilled and then inserted into the extensions of the motor which holds the dummy roller in the front.

Fug.5 Dummy roller

Circular holes have been carved onto the sides of the dummy roller and a pair of 6200z bearings has been placed as visible in the picture to provide free rotation. As we can see, ridges have been carved carefully so that the bicycle chain can easily fit within the space created hence connecting the front wheel with the rear one. The bicycle chain that we used is the traditional Bangladesh rickshaw chain. Its strength and durability is perfect for the waiter bot project. Working with it can be monotonous but it does have its advantages. With this chain we can construct various types of wheel structure, pulleys if required and more. In the picture below, we can see how the treads are placed and also what spacing is maintained. The plates between the two chains acts as extra support for the chain track and helps to provide more stability as the entire structure performs necessary rotation. The plates are actually cut out from steel sheet, bent to accurate measurement then drilled on the sides for the screws to be used. The steel supports are of at least 1mm thick and roughly 4-5 inches wide giving it further stability and shape. Below is a picture which shows how we have attached the plate supports using screws. A total of 12 plate supports has been used to construct the chain track and here is how the entire track looked like after construction.

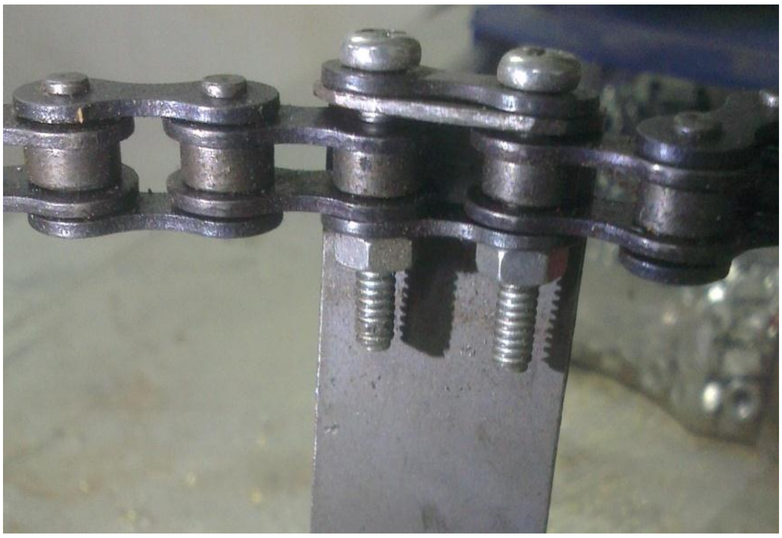


Fig. 7 Supports attached to chain track with screws

Each of these steel supports were connected together to make a complete chain track system which we were capable of attaching around the HUB motor and the dummy roller in front. The next picture shows the work implemented and what the chain track system looked like.

Fig. 8 Complete chain track

From this picture we can see that the plate spacing is purposely increased in order to ensure fluidity of the rotational motion. If this was done exactly like the software model then the fluidity to the robots motion would not have guaranteed. We do not want unnecessary problems which involve static situations during the rear wheels rotation and hence lead to the reconstruction of the chain track. Similarly, a pair of these chain tracks have been constructed and put into use. Here is a three dimensional model of one side of the bicycle chain track we have built. In this partially completed snapshot we can see the two rollers (front and back), the chains and the some of the supports. After constructing a pair of these chain tracks, they have been attached with two steel bars welded into proper positions thus creating the lower part of the waiter-bot. this serves as the legs or wheels for the robot with which it can move about according to necessary requirements. A picture of the complete construction of the lower part of the robot is shown in the next page.

Fig. 10 Lower part of the robot

In the picture above, we can see the overall locomotion system that the robot will use to move about. This will be done with the help of microcontrollers which will be programmed by arduino and will also be powered by a DC power supply. The details of HUB motor and its specifications will be discussed later. A picture of the lower part of the robot viewed from another angle.

Fig. 11 Lower part of the robot viewed from another angle

3.1.2 Robotic Arm and Linear Actuators:

According to our software design we found that two 12 volt actuators are totally compatible for our desired robot. Our initial idea was to make two separate hands for serving the meal by grabbing the tray on which the meal is placed. Later we placed the two actuators in two sides of the robot on two supports attached to the lower part of the robot as shown below. For serving the meal, we decided to do it in a linear motion according to the respective height of the table. With the help of actuators we will be able to perform this task with precision and ease. We could have made the body of the robot to rotate and face the table before serving the meal but based on the environment that we have set up, rotating the body may involuntarily come in contact with someone as they pass by and may result in an accident.

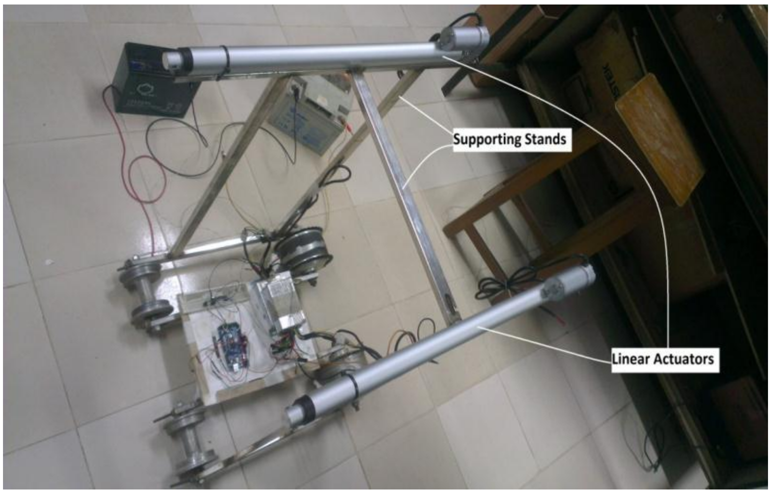


Fig. 12 Upper part of the robot

3.2 Problems during implementation and its Solution:

After the implementation of the chain track as discussed before, we have realized that the chain gets displaced during the rotation of the motor. It totally comes off from the grooving constructed on the HUB motor using spockets. To keep this design by overcoming this problem requires high level of precision and accuracy in building the chain track itself. Another problem that we have faced is that the robot vibrates too much during motion if the chain tracks are used. This may cause the tray on which the meal is served to fall off as it travels on its path. This particular problem can stand as a huge risk and will undermine the projects standard and quality. As a result we tried to find a solution to this problem which is that the robot has to have a smooth ride all the way. Hence if so, the chain track has to be removed and put out of consideration and a new system has to be implemented. The only best possible solution is that if we could put a thick rubber material of good quality and durability over the HUB motor then the robot will be granted its smooth ride and the possibility of the meal to fall off would have been diminished. Based on the assumptions made, we now had to decide what rubber material to use and were able to come to a conclusion that the rubber used for CNG tires is a promising option. This particular rubber material is easy to manipulate and easy to implement and is a fitting solution to our problem. Lastly, after the physical design was changed the robot performed smoother and problem free. Below is a picture of the HUB motor after the chain was removed.

Fig. 13 HUB motor with CNG rubber tire attached

The next problem was to automate the switching mechanism of the robotic arm. It was clear to us that switching a single actuator is better and easier than switching two actuators at once. It requires less space and thus provides less complexity to the robot. It also enables us to maintain a systematic balance to the robot according to its specifications. As a result, we decided to change the upper part of the robot and implement a single actuator placed right in the middle of the supporting stands.

Fig. 14 Single Linear Actuator with supporting stands

Chapter 4: Circuit Design and Set-up:

4.1 Arduino Mega:

The main brain of the waiter-bot is the microcontroller which was used to help with the implementation of the entire system architecture. Arduino Mega 2560 is a microcontroller which has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. [3]

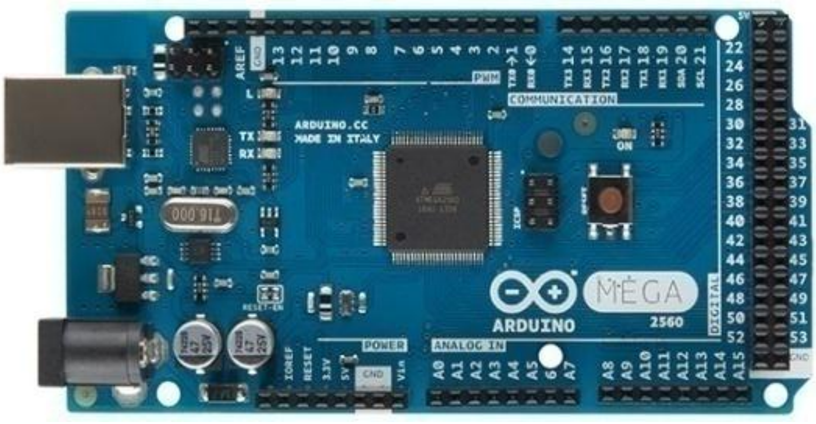


Fig. 15 Arduino Mega

The reason why we have chosen to use the arduino mega is because it has a huge number of pins that can be utilized according to our requirements. Since our system architecture requires a large number of pins, the arduino mega is a suitable option for this project. t has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 hardware serial ports, a 16MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller. Simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. [4] The software and code sequence will be discussed in the next chapter.

4.2 Keypad Matrix:

The 3X4 foil matrix keypad has 12 buttons, arranged in a telephone-line 3x4 grid. It's made of a thin, flexible membrane material with an adhesive backing so you can attach it to nearly anything. The keys are connected into a matrix, so you only need 7 microcontroller pins (3-columns and 4-rows) to scan through the pad. A picture of the keypad is provided below. [2] We have tried to maintain the simplicity to the work of this project by reducing the number of tables to four tables in our environment set-up depending on the work space we were provided. Never the less, the waiter-bot is able to serve to as many tables as necessary but according to our specifications. Since we were working with a small number of number of tables, there was no need to use 4X4 or any other keypad matrices. If so, then the coding would have been longer and difficult and hence would create complexity to the software implementation.

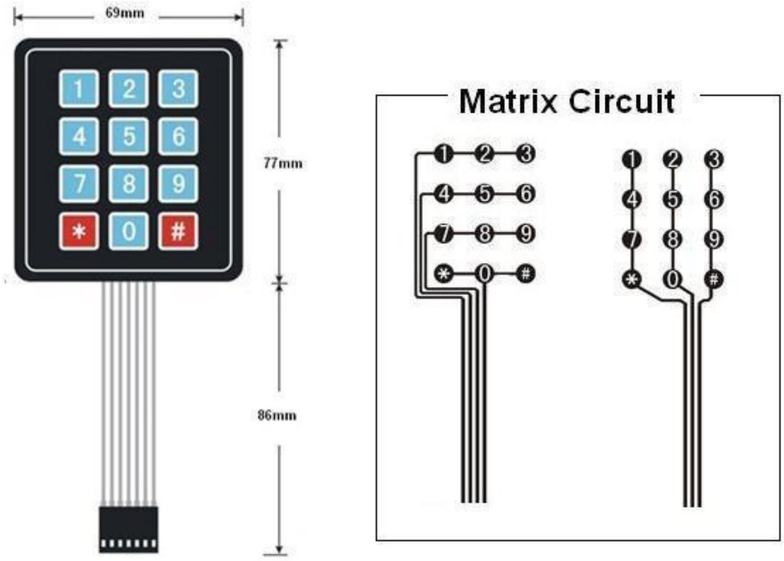


Fig. 16 Keypad Matrix

We have used the buttons to indicate the table numbers and will indicate the destination to the robot. To make this happen proper coding is necessary so that the robot finds out the exact route for the final destination. Before doing so, we need to know how to connect the keypad with the arduino and also figure out the coding necessary for making the keypad function properly. Below is an attached picture of how the connections are given in order to connect the keypad with the arduino. In the picture below, the green lines indicate the rows of the keypad and the orange lines indicate the columns of the keypad matrix.

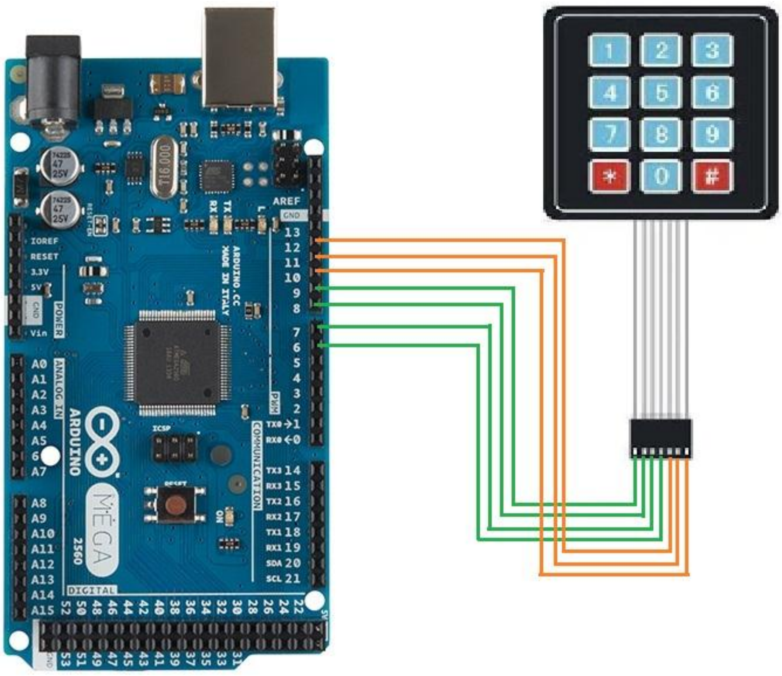


Fig. 17 Circuit connection of the Keypad Matrix

4.3 IR Sensor:

The waiter-bot is a basic path finder which seeks out its destination according to the line that it is meant to follow. To detect this line a particular array of IR sensors are used. The IR sensor consists of five pairs of infrared emitters and receivers which detect what color of line is beneath the sensor. The receiver passes the readings taken into the arduino as analog signals. The arduino then interprets those signals and decides which line to follow. Later, it sends logic signals or PWM signals to the motor controller units to drive the motor accordingly. The relationship between the IR sensor array and the motor via the arduino is given below.

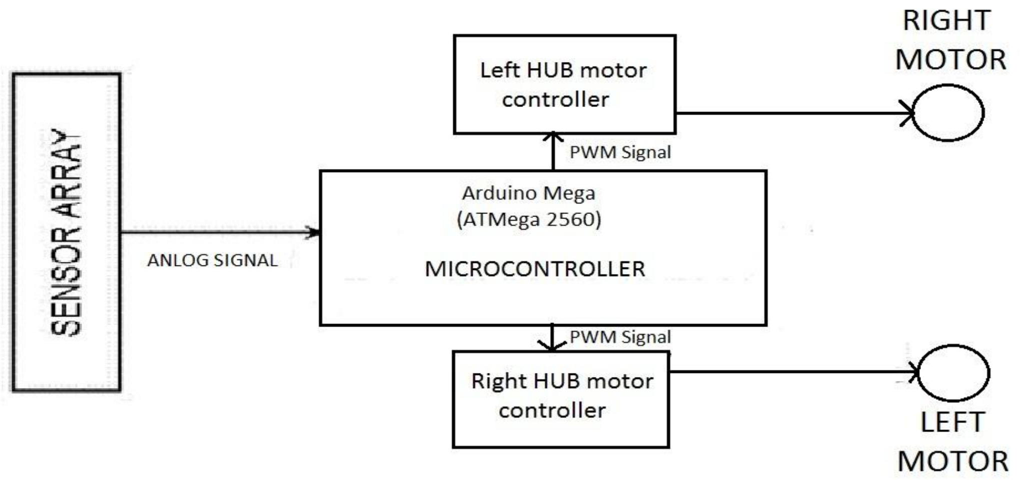


Fig. 18 Relationship between sensor array and motor

In our particular robot, after a series of experimentation we were able to find out that the sensors are able to detect the color white within a value above 45. This means The IR sensor will give the value equal or above 45 when it will find white lines and below 45 when it detects black lines. We used the and operation (&&) in the code all the values has be equal or above 45.The IR sensor sends the values in the arduino which will recognize the line as black or white and give instructions whether to go forward or stop. The coding of the complete line following process and other details will be discussed in the next chapter. Let us now see the connections of the IR sensor Array with the Arduino.

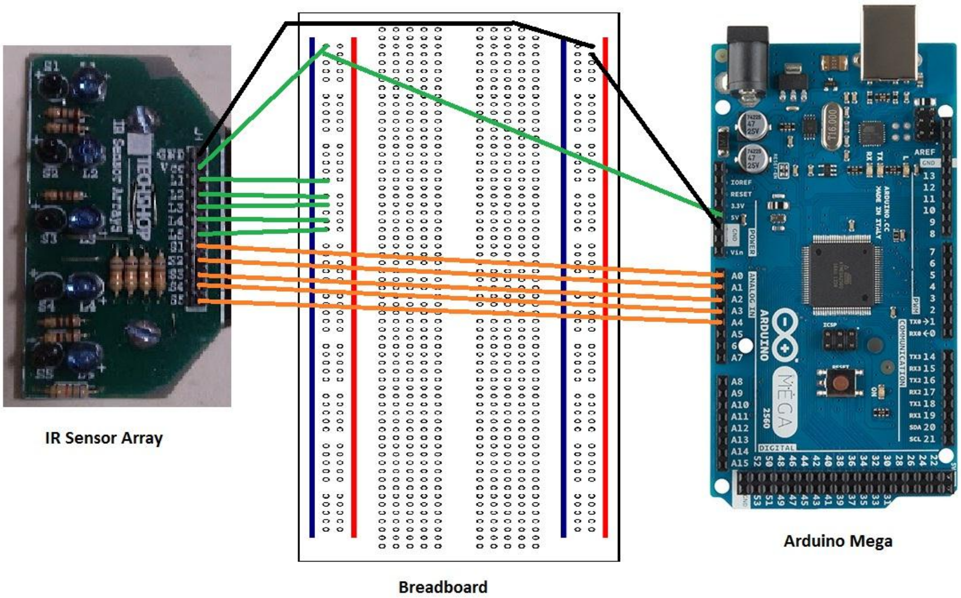


Fig. 19 IR Sensor Array with Arduino Mega

It is simply visible from the picture above, that the IR sensor array has a voltage pin (Vcc) to activate the board and a ground pin (GND) to make the ground connection. These two pins are connected to the arduino 5 volt and the arduino ground via the breadboard as shown. Next, we can see there are a series of emitter and receiver pins labeled L1 to L4 and S1 to S4 respectively. The emitter pins are connected to the arduino 5 volt on the breadboard and the receiver pins are connected to the analog pins of the arduino as shown. The data of the receivers received by the arduino is actually an analog data, the reason why they are connected to the analog pins of the arduino. In general, any analog reading if is needed to be taken then they should be connected to the analog pins of the arduino. With the help of the sensor array the arduino is able to decide which route to take according to its code and can then signals the motor controllers to drive the HUB motors accordingly.

4.4 Ultrasonic Sensor:

Ultrasonic distance sensor determines the distance to an object by measuring the time taken by the sound to reflect back from that object. The frequency of the sound is somewhere in the range of ultrasound, this ensures more concentrated direction of the sound wave because sound at higher frequency dissipates less in the environment. A typical ultrasonic distance sensor consists of two membranes. One membrane produces sound, another catches reflected echo. Basically they are speaker and microphone. The sound generator generates short ultrasonic impulses and triggers the timer. Second membrane registers the arrival of the sound impulse and stops the timer. From the timers time it is possible to calculate the distance traveled by the sound. The distance to the object is half of the distance traveled by the sound wave. [5]

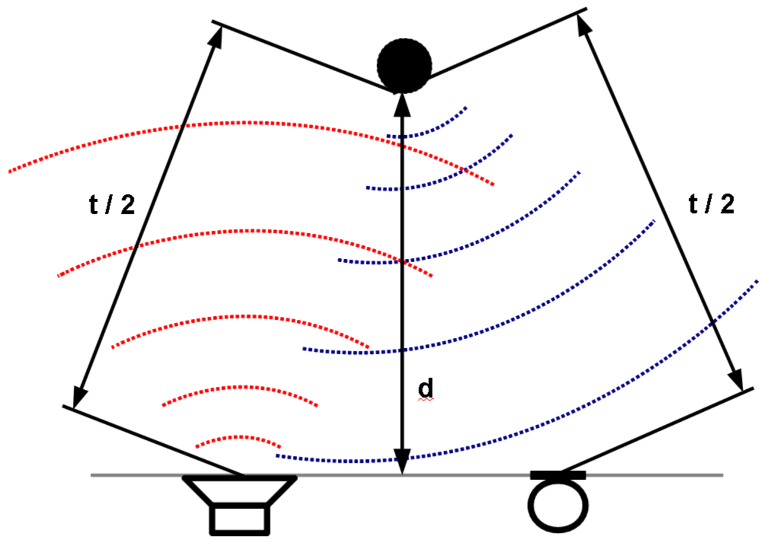


Fig. 20 Ultrasonic Working Principle [5]

In the picture above, a clear working principle of the ultrasonic sensor is shown. One membrane is the transmitter of ultrasound and the other is the receiver. These two units work together in making the ultrasonic sensor work in its own efficiency. The time taken for the ultrasonic wave to transmit and reach the object is half the time it takes to travel and come back to the receiver membrane. Using ultrasonic sensors in any robot allows the users the cutting edge technology to detect and avoid obstacles accordingly. Let us see how to we have connected the ultrasonic sensor to the arduino.

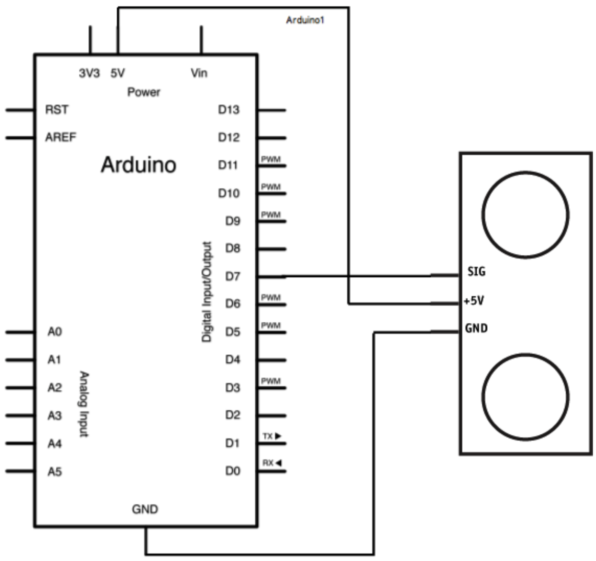


Fig. 21 Ultrasonic Sensor Connection to the arduino [6]

The ultrasonic sensor has three pins. One is the 5 volt pin (Vcc) to activate the ultrasonic board and the other is the ground (GND). These two pins are connected to the arduino 5 volt and the ground pin respectively. The third pin is the signal pin or else known as the data pin. This pin is connected to any one of the digital input/output pins as necessary. This is the pin through which the ultrasonic sensor passes the transmitted and received data to the arduino and works according to the code implemented. Based on the working principle, the implementation of the ultrasonic sensor provides a unique intelligence to the robot. For example, the robot will be capable of detecting anyone or any object in front within a certain distance and will be able to stop in its tracks according to its code specifications.

4.5 Servo Motor and Linear Actuator:

There are some special types of application of electrical motor where rotation of the motor is required for just a certain angle not continuously for long period of time. For these applications some special types of motor are required with some special arrangement which makes the motor to rotate a certain angle for a given electrical input (signal). For this purpose servo motor comes into picture. This is normally a simple DC motor which is controlled for specific angular rotation with help of additional servomechanism. The main reason behind using a servo is that it provides angular precision, i.e. it will only rotate as much we want and then stop and wait for next signal to take further action. [5]

Fig. 20 Servo Motor

Servo motor has three pins, 5 volt, ground and a data pin which connects to the arduino. Based on the code uploaded to the arduino the servo motor rotates accordingly. We have taken this rotation of the servo motor panels and put it into our benefit. The actuator which was finally chosen to be used was a single 24 volt linear actuator. Its dimensions and size to weight ratio is perfectly suitable for the implementation of the robotic arm of the waiter-bot. Now, the critical issue is the switching mechanism of the robotic arm during serving of the meal. The solution to this particular problem in fact is totally mechanical. Below is a picture provided of how this mechanical switching has been implemented.

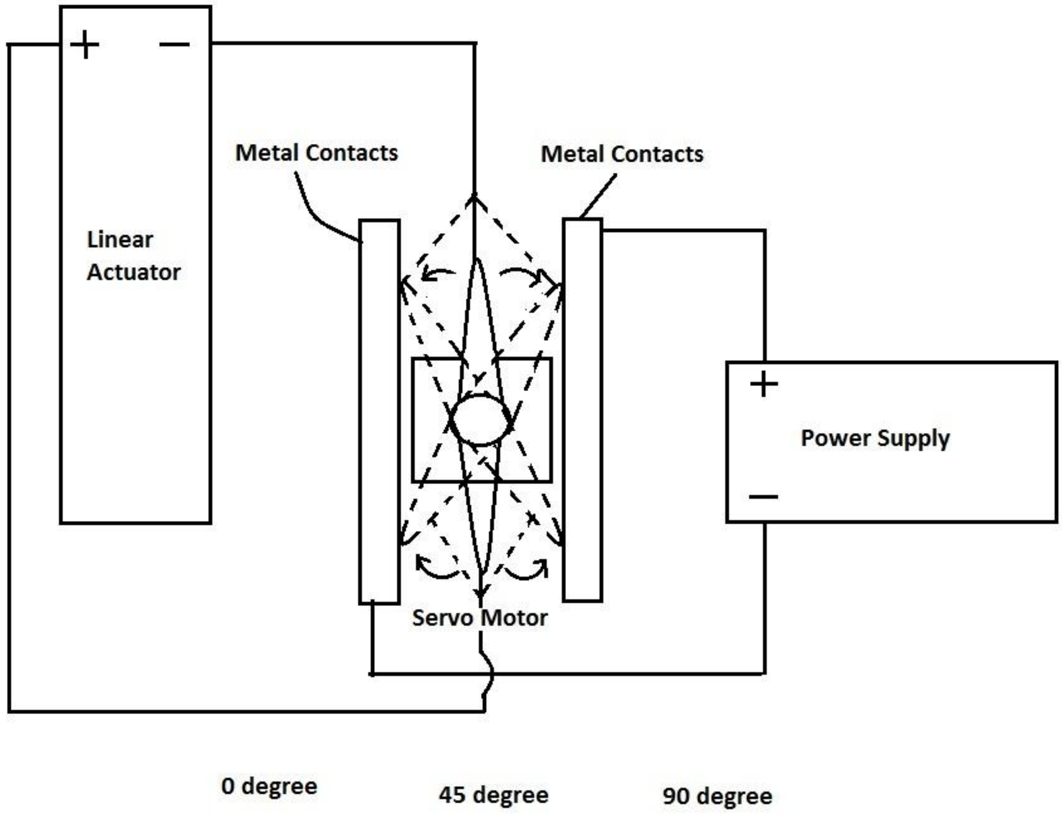


Fig. 21 Switching Mechanism

As we can see in the picture above, two metal plates were placed on the two sides of the servo motor. The terminals from the power source were then connected to the metal plates. As a result a constant supply was made available at all times for the actuator to connect to. Now, we had to build a switching mechanism for the terminals of the actuator which will allow it to change direction according to the timing of the servo motor rotation. The terminals of the actuator are attached to the panels of the servo motor. When the motor rotates to zero degree and connects to the metal plates, the actuator receives power from the source and extends and hence the process for serving is completed. Next, when the servo rotates to its 90 degree position, the actuator gets connected in reverse to the terminals of the power source and therefore comes back to its starting position in its given time. Once back in its starring position the servo rotates its panels to its 45 degree position which is also known as the idle position of the servo motor. Is this state the connection to the linear actuator is severed and in doing so the actuator is disabled.

4.6 HUB motor and HUB motor Controller:

Here in this section we are going to discuss the circuit connections made for setting up the HUB motor for the movement of the robot. To drive the HUB motor the HUB motor controller is compulsory. The HUB motor which we have used has a current of 15A to 17A and a power output rating of 250W to 350W. These are just specifications of the motor. But the actual power output and the current value depends on how much load is being applied to the motor. For example, weight of the robot including the entire body frame applies a certain load resistance to the HUB motor. In such cases, the motor will require more charge and hence will draw out more current from the power source. For our robot to function according to limited requirements, we are going to use only the hall sensor pins and the motor phase pins from the HUB motor and the power supply and the throttle control pins from the HUB motor controller. Using the hall sensor pins and the motor phase pins the HUB motor will receive a set of instructions from the throttle control pins of the HUB motor controller based on which the motor will rotate. The connection is as follows. The color of the motor phase pins are yellow, blue and green which will be connected to the same color combination of pins with the motor controller. Next, the hall sensor pins of the HUB motor consists of five pins of colors red, black (5 volt and ground respectively), yellow, green and blue will be connected to the same color combination of that of the motor controller. Later, to provide the system with a power supply we have to connect it to the power pins (red and black) which is situated in the motor controller not the HUB motor. This is the reason why using a HUB motor require a Motor controller unit to make it run.

4.7 Battery Specifications:

The most important issue to be clear about is what amount of power supply the total system needs and how we can set-up the power supply accordingly. Based on the necessities of the electronic elements used for the robot, a total of 24 volts is sufficient enough to run the electrical and electronic components of the system. The working voltage required for a single HUB motor is 24 volts. Hence for two motors a total of

48 volts is necessary. Again, the actuator that we have used is a linear actuator with a voltage rating of 24 volts. Therefore an additional 24 volts is coming into play. Connecting all these elements individually is a very wrong way to make an implementation. The problem with setting up the power connection individually will cost us more space and also pose a weight management problem. We also have to keep the robot body balance in mind. Maintaining the balance of the robot with an excess weight is more difficult than maintaining the balance within the weight specifications. If we notice properly, every running element requires 24 volts each. So, we can connect all these elements to one power source of 24 volts in parallel and hence reduce the risks of long term disasters due to excess weight. Then again, a single 24 volt DC battery is still bigger in size and a little heavier. The space that we were able to make in between the two HUB motor systems does not allow us enough space to place the single 24 volt battery. Below a picture of how the power connections were applied to the HUB motors and the actuator is given.

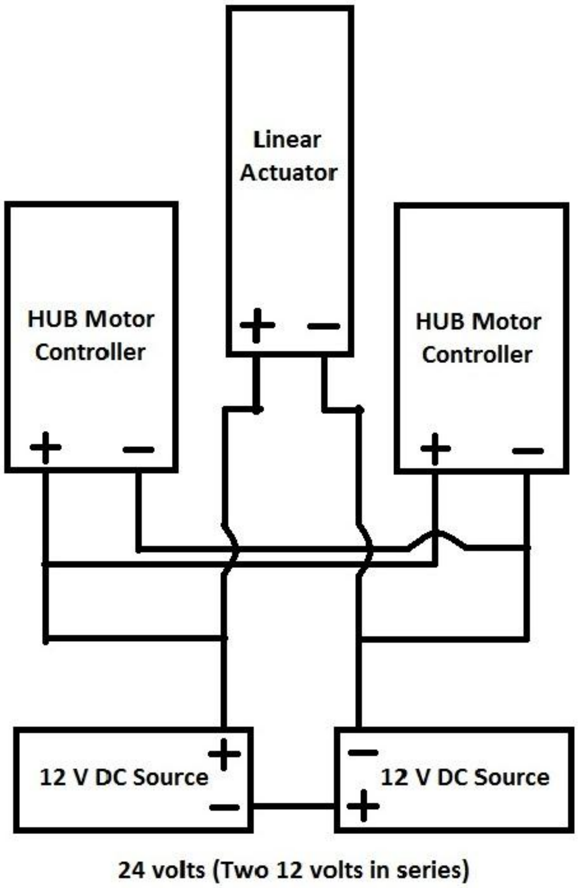


Fig. 22 Parallel Connection

As we can see, in Fig. 21, two separate DC voltage sources of 12 volts have been made series and then the connections were given accordingly. Since both the HUB motor controller units require same voltage they have been connected in parallel as well. This provides us with the liberty to use one source for two controller units. Now, the same source is also used for providing connection to the linear actuator via servo motor as discussed before. Thus a simple power set-up is produced reducing the requirement of space and the possibility of risks during the robots utilization.

4.8 Complete Circuit set-up:

In the previous topics we have discussed about the circuits of each element individually. Now it is time to assemble them all together in a systematic manner. In this topic we will see how each of the elements has been placed in its proper position which was thought to be suitable according to our requirements. The IR sensor array is placed in between the two front dummy rollers screwed under the platform at about 4.5 to 5 inches above the ground. The two HUB motor controllers were placed on the two sides just in front of the HUB motors. The power source were placed right behind the HUB motor controllers side by side in order to balance out the weight evenly on the surface on which the entire circuit was positioned. The space in between the HUB motor controllers was used to position the breadboard connections and the arduino as a sort of a centre point of the entire circuit set-up. This is what the electrical and electronic system looks like in reality.

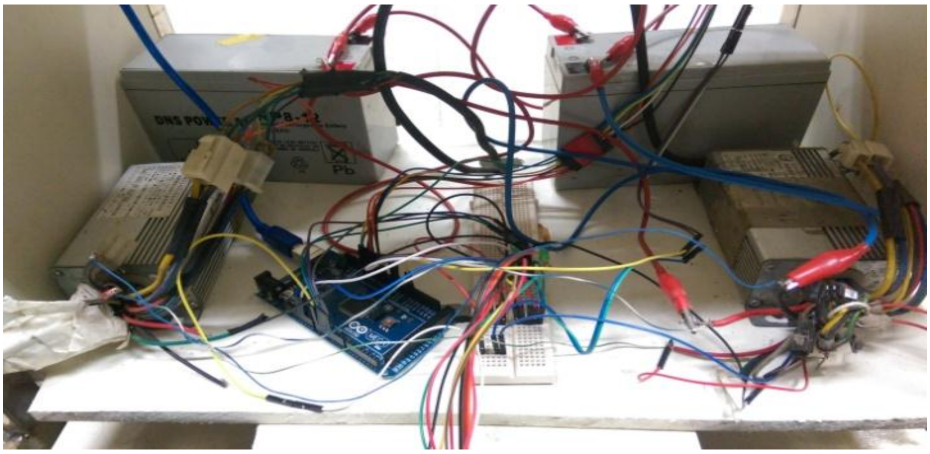
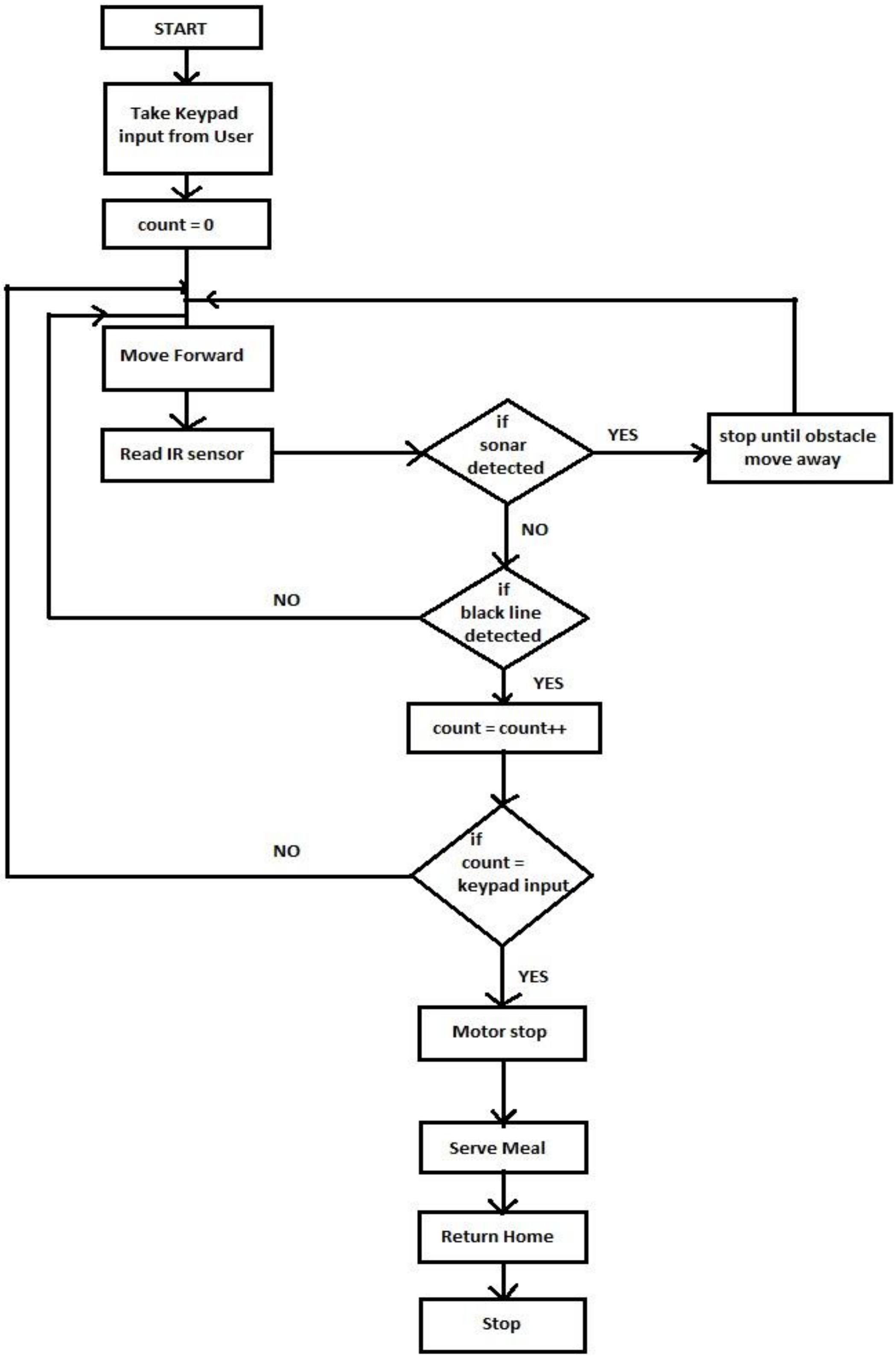


Fig. 24 Complete Circuit Set-up

Chapter 5: Code sequence:

5.1 Flowchart:



Chapter 6: Result Analysis:

In this chapter we will see precisely how the waiter-bot performed and how much success rate it has achieved. Based on the number of trials executed from the beginning of the project implementation, a record of each success and each failure were noted. With the help of this data we were able to create a bar chart which shows us exactly how many times the project was a success and how many times the project was a failure out of a total number of trials. The bar chart is given below.

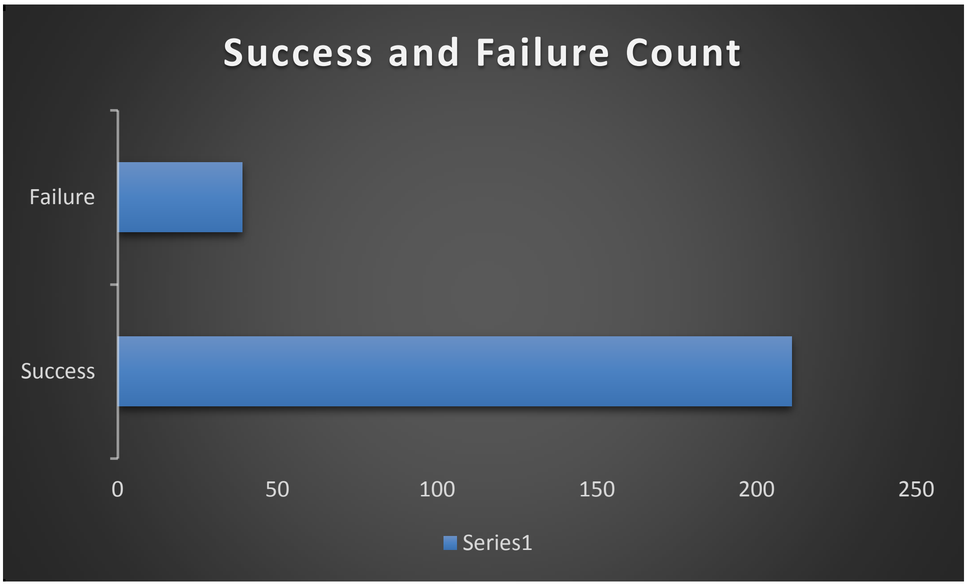


Fig. 25 Success and Failure Count

As you can see from the bar chart, it is clear that out of a trail of 250 times, we were able to achieve a success of 211 times and have also confronted with a failure of 39 times. To be more elaborate, the first 11 to 16 trails were a continuous failure. This was because at the beginning of the project we were unable to make the robot detect the line itself. The IR sensor values used in the code were not right and the distance between the IR sensor array and the ground was not proper. This particular issue was solved after a long monotonous but careful process of tuning. By now the robot was somewhat capable of detecting white line and black line with quite legitimate accuracy. None the less, the robot was not able to stop at the right place at the right time. This is because the motor speed was initially high and it took the robot some time to come to a complete stop. By the time it did, it used to stop just a bit front of the respective table. This was due to inertia of motion of the robot as the robot had a considerable amount of weight. Thus the next few trials were also mixed with a bit of success but more failures. As a result the motor speed had to be calibrated according to our necessity and the robot was tuned to perfection. Finally, towards the end of the trials the robot functioned flawlessly without the requirement of further tuning. A pie chart of the total percentage of success and the total percentage of failures is provided below.

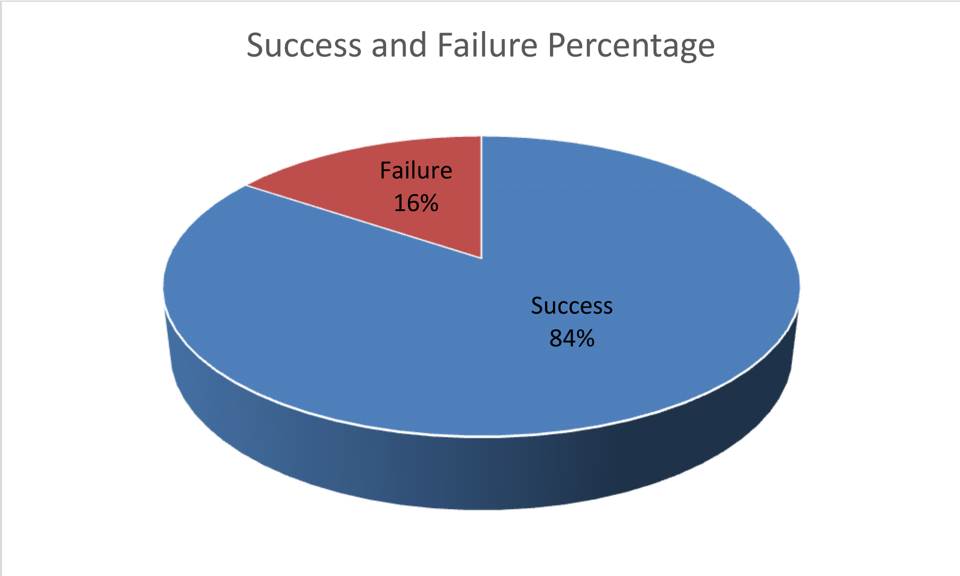


Fig. 26 Success and Failure Percentage

As you can see, the pie chart above gives us an estimate of the precision and accuracy of the waiter-bot. after extreme hard work and relentless trial and error procedures we were successfully able to generate a success rate of 84% with the waiter-bot.

Chapter 7: Conclusion and future scope:

The introduction of the waiter-bot in the markets of Bangladesh will bring about a drastic change in the standard of our country to its neighboring countries. Not only that, it will also get a worldwide technological recognition. In the particular field and business, this project is the first and may have a high probability to become an influential aspect towards an economic boost. Bangladesh business is a rapid expanding market in its own way. If, with a strong patience, this project is tested on the field directly for a couple of months and the results on the market situation and the business status is analyzed, then an even better conclusion may be driven from its effects on marketing strategy. It is guaranteed that a particular restaurant with an automated staff will attract a lot of customers within its domain. People will seem to simply prefer that particular restaurant just to see the automated staff in action. With hard work and creativity, a particular restaurant such as this may turn into a tourist spot for many foreigners. A lot of side business may take a hold in the market due to the influence of automated restaurants in the near future. If so as more and more restaurants become automated the basic revenue from each of these restaurants will surely provide an extreme economic boost to the nation. Besides the business and marketing advantages, there is more to the waiter-bot than what it seems like. The use of waiter-bot will reduce human effort and create a comfortable lifestyle for the residents of Bangladesh. Never the less, it will ignite the vision of complete automation in the near future if the youngsters of this generation and the next are introduced to this particular simple yet new technological implementation.

Chapter 8: Reference:

[1] "Restaurant in China Employs Robotic Wait Staff & Chefs,"

http://www.designnews.com/author.asp?section_id=1386&doc_id=258859&dfpPParams=ind_182,aid_258859&dfpLayout=blog&dfpPParams=ind_182,aid_258859&dfpLayout=blog , Feb 5, 2015

[2] "Adafruit Industries, Unique & fun DIY electronics and kits,"

http://www.adafruit.com/products/419, Jan 2, 2005

[3] "Pololu – ArduinoMega 2560 R3,"

https://www.pololu.com/product/1699/resources, Dec 20, 2013

[4] "Arduino MEGA,"

http://www.mantech.co.za/datasheets/products/A000047.pdf, Mar 15, 2015

[5] "Ultrasonic distance sensor [Robotic & Microcontroller Educational Knowledgepage - Network of Excellence],"

http://home.roboticlab.eu/en/examples/sensor/ultrasonic_distance, Jan 6, 2015

[6] "Arduino - Ultrasonic Distance Sensor,"

http://wiki.t-o-f.info/Arduino/CapteurDeDistanceUltrasonique, Mar 25, 2015

Appendix:

Code implementation using arduino software:

```
#include <Keypad.h>

#include <Servo.h>


//initializing servo

Servo myservo;

//initializing all variable

int serv = 0;

int recv = 90;

int idle = 45;


//initializing rows and columns for keyboard

const byte ROWS = 4;

const byte COLS = 3;


//initializing other variables

int sensorArray [5];

int tableNo;

int currentTable=0;

int totalTable = 4;

int rightspeed = 200;

int leftspeed = 120;


//this is the value that changes the light sensitivity

int distance =45;
```

```
boolean ok = false;

int motorLeft = 3;

int motorRight=2;


//defining keypad

char keys[ROWS][COLS] = {

{'1','2','3'},

{'4','5','6'},

{'7','8','9'},

{'*','0','#'}

};


byte rowPins[ROWS] = {9, 8, 7, 6};


byte colPins[COLS] = {12, 11, 10};


Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );




void setup() {

  Serial.begin(9600);

  stp();

  currentTable=0;

   myservo.attach(14);  //stablishing connection to the servo

  myservo.write(idle); //setting it to a central position
```

```
}




void loop() {
getKeyBoard();
 if(ok){
   getIrValue();
 }
}

void getKeyBoard(){
  //get and store the key pressed into local variable
    char c =keypad.getKey();

    if(c>0){
    tableNo = c-0;
    //map/transform the value to real integer number. As the input is a form of ASCII
    tableNo = map(tableNo,48,57,0,9);
    Serial.println(tableNo);
    ok=true;
    }
}
void getIrValue(){
  int sensorArray [5];
```

```
  sensorArray[0] = analogRead(A0);

sensorArray[1] = analogRead(A1);

sensorArray[2] = analogRead(A2);

sensorArray[3] = analogRead(A3);

sensorArray[4] = analogRead(A4);
  Serial.print("First : ");
  Serial.println(sensorArray[0]);
  Serial.print("Second : ");
  Serial.println(sensorArray[1]);
  Serial.print("Third : ");
  Serial.println(sensorArray[2]);
  Serial.print("Fourth : ");
  Serial.println(sensorArray[3]);
  Serial.print("Fifth : ");
  Serial.println(sensorArray[4]);
  delay(250);
  checkCondition(sensorArray);
}

void checkCondition(int sensorArray[]){
```

```
  //for around 5 inches and off-white surface 40 is more than enough. Changing this value will increases or
decreases the range of reflection
  if((sensorArray[0]<=distance)&&
(sensorArray[1]<=distance)&&(sensorArray[2]<=distance)&&(sensorArray[3]<=distance)&&(sensorArra
y[4]<=distance)){

    if(currentTable<=tableNo){
      if(currentTable==tableNo){
      Serial.println("Serving Food");
      stp();
      currentTable =0;
      serveFood();
      returnHome();
      ok = false;
    }else{
      go();
     Serial.println("Achi");
  // delay(1000);
    }
  }

  }else{
    Serial.println("nai");
  //  delay(1000);
    currentTable++;
  }
```

```
 Serial.print("current: ");
 Serial.println(currentTable);
 Serial.print("table no: ");
 Serial.println(tableNo);
// delay(500);
}

void go(){
  Serial.println("go");
 analogWrite(motorLeft,rightspeed);
    analogWrite(motorRight,leftspeed);
}

void stp(){
 analogWrite(motorLeft,0);
   analogWrite(motorRight,0);
   Serial.println("Stopped");
}

void returnHome(){
 Serial.println("Going Home");
}

void serveFood(){
myservo.write(serv);
   delay(17000);
```

```
myservo.write(recv);
    delay(17000);
   myservo.write(idle);
}
```