SZABIST

Project Advisor

Sir Abdul Basit Aftab

Ms. Tanzila Younas

Ms. Nasreen

Submitted by

Syed Fareed Alam Nizami 2045123 -6A

Syed Nizam Alam Nizami 2045126 -6A

Syed Ali Ahmed 2045157– 6A

Department of Mechatronic Engineering Shaheed Zulfikar Ali Bhutto Institute of Science and Technology

Abstract

The design and construction of an autonomous ball-shooting robot are presented in this study. The robot follows a predetermined path; it chooses the target placed at predetermined intervals and heights along the way; and it shoots table tennis ball at a desired target. Robot is outfitted with various electrical and sensor components that are all interfaced to a microcontroller in order to navigate through the black reflective lines (Arduino Mega). The linearization of the Sharp IR sensor is the main foundation of the robot's navigation algorithm. The robot then incorporates this knowledge into its control algorithm, efficiently keeping it in the middle of the corridor and at turns throughout motion without encountering any conflicts. These robots are used in industrial pick-and-place activities.

Table of Contents

Chapter 1: Introduction

1.1. Overview

For this project, an autonomous ball shooting robot was to be created. It would be a motor driven robot with wheels, that would move on a predetermined path and shoot balls at the target. In its most basic form, it is a line following robot that uses infrared sensors to follow a black line, after it detects a target, it would shoot a ball out by the help of a servo motor and shooting mechanism. It was specified that the size of the robot be kept up till 10”x10”. The aim was to create a robot that would complete the task as quickly as possible completing the pathway in as little time as possible and for each correct ball properly shoot at the target, student received points.

1.2. Background

The project consists of an autonomous ball-shooting robot which can be considered a small- scale prototype of an industry level autonomous sorting robot. As the world moves into the future, it is seen that every operation needs to be automated to improve efficiency and improve rate of production. Machines are now made to perform entire tasks by themselves and are expected to decide what needs to happen. This project was given to allow students to realize how machines can be automated, how processes can be automated. It is a test of design and efficiency. What technique a student employs to perform the task as quickly as possible. In warehouses, employees need to pick boxes and place them in their designated position so that they can be shipped to the right place. A similar concept is employed with the autonomous ball shooting robot. It places balls into their designated position and does this at three different locations. This can be scaled to be a larger project used in the real industry

1.3. Motivation

The motivation for this project was the need of automation in Pakistani industries. Pakistan, despite its large labor population faces production decline, as the world is moving towards automated process, Pakistan has been left behind. To allow Pakistan to become competitive, it requires automation on a level that is cheap but scalable to a very large degree. Since lots of downsizing would be required to employ an automaton in place, the prices for one need to be very low for it to be an attractive deal. Students had to use their minds to create an attractive solution to the problem that could efficiently automate a process.

1.4. Objectives

The objective of this project was for the students to employ all concepts of their studies. All areas of studies were tested including computer programming, electrical, mechanical, and designing in all its forms. Students were supposed to deal with this project as a problem and come up with a creative solution to solve the issue. This is where their designing skills for tested. The overall goal of the project was to create a robot that could perform the task as quickly as possible with as much accuracy as possible.

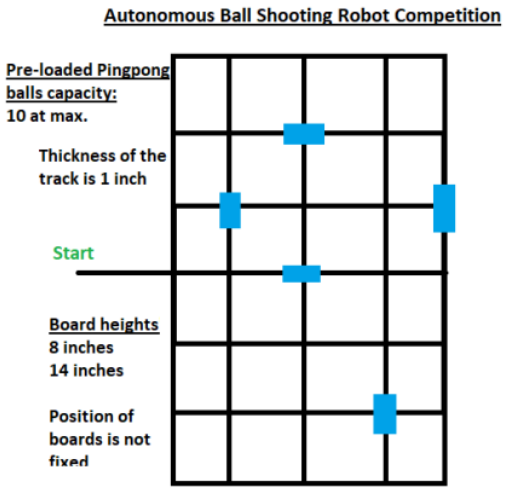
1.5. Application

Since this project is small scale, if it were to be scaled up for different operations, it can be applied to different parts of many industries. Sorting robots have a reclining tray and a barcode reader, so they can sort packages and arrange them in the correct outgoing line. The operator places the items in the robot at the pick station, which scans the label, processes the information, and transports the parcels to the dispatch area. Mezzanine-operated robots are another type of sorting robot. It contains goods entering channels on the top level and exit ramps leading to the dispatch area. Workers load the parcels into the robots, which receive, sort, and transport the merchandise to the appropriate exit.

Chapter 2: Experimentation

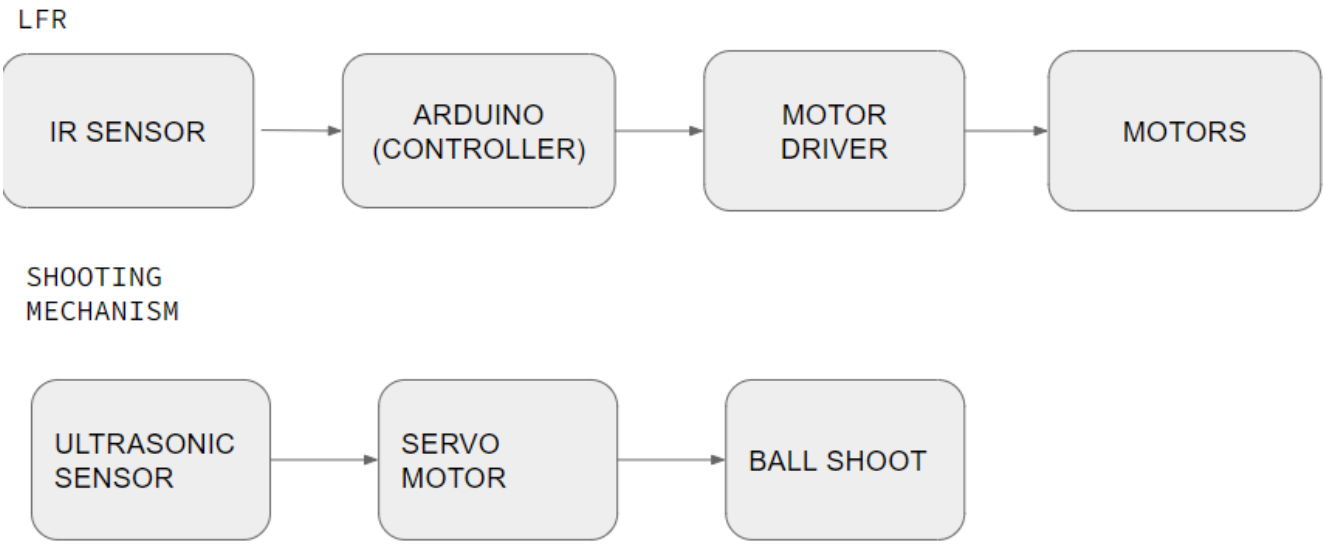
2.1. Design of Experiment

Autonomous Line following Robot is proposed in this study. A certain track is chosen, and the robot is required to proceed in that direction using data from the sensors. Sensors such as ultrasonic, proximity, and infrared are utilized and affixed to the robot's front dimensions. All sensor data is supplied into the controller, which compares all of the values and outputs movement of the robot and gating channels. However; different HC values can be utilized to improve the robot's efficiency so that it can perform better on the track.



2.1.1. Block Diagram

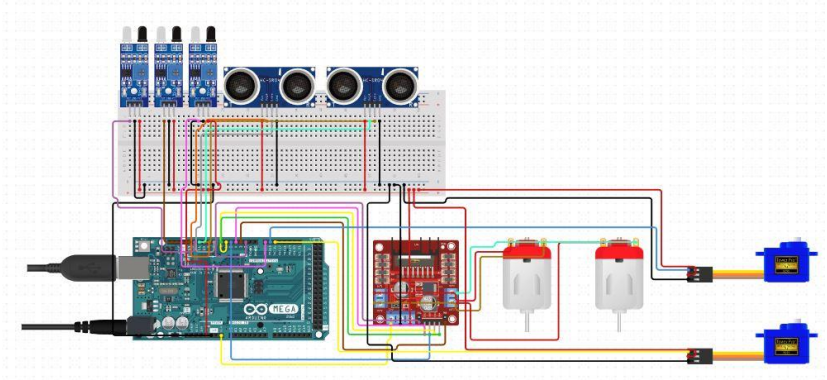
Following is the block diagram of our project.



2.2. Experimental Setup

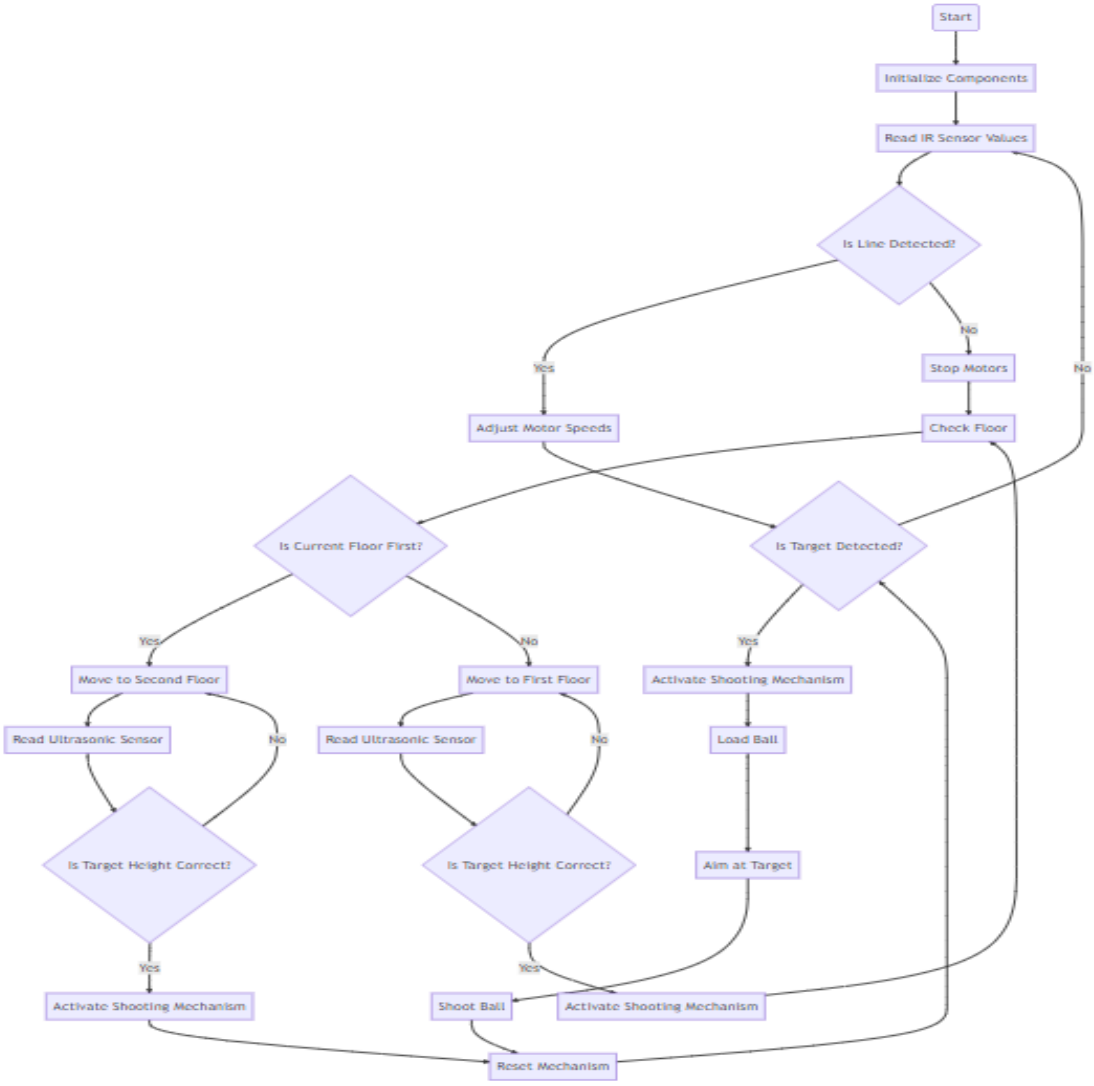
2.2.1. Schematic Diagram

Following is the schematic diagram of our project.



2.2.2. Flowchart

Following is the flowchart of our projects working.



2.2.3. The algorithm

This project was designed to do the multi-tasking. 1st task that it should perform is to follow any path with IR sensor modules or with hard coding. 2nd task is to detect the target. 3rd task is to operate the servo motor and turning dc motors with wheels on. And the last task is to shoot the ball. These are the 4 tasks to be perform.

2.3. Equipment and Materials

2.3.1. Materials

There were many materials used in making this project. For the base and structure, we first decided to use wood and started our project with wood. But when we completed our base for the project, there we found our first complication. Weight of wood was too heavy for the motors to bear and it was just the base and there were many other things to put so it would have crashed. So, we dismantle our base. After that we thought of using acrylic sheet.

Acrylic was light in weight. So, we use 6mm acrylic sheet which makes our base was strong enough to bear the weight of the whole project. There are three floors, we used acrylic sheet for the base supported by rectangular acrylic rods. We 3D printed the brackets to support the structure and we also 3D printed ball holders and shooter.

2.3.2. Equipment

- Other than these we also used the following equipment:
- Screws and nuts for the brackets and acrylic rod fixing.
- Glue gun for fixing and placing objects in the project.
- Gmsa glue for acrylic to fix smoothly and firmly.
- Double sided tape to place and fix the electronic parts.

2.4. Tools and Platforms

2.4.1. Tools

Our project was a multitasking robot. For it to do multitask, we had to make 3 floors in our project. We also had the restriction of base size. To gain extra marks we had to design small size base but the dimensions of the other floors can be as large as we design.

So, to assemble our project we had to use many tools like:

- Screwdriver to tighten the screws used in wood.
- Wrench to tight the bolt.
- Paper cutter to cut the floor design.
- Laser cutting machine to cut the acrylic pieces.
- Small size screw drivers for electronic components.

2.4.2. Platforms

To complete our project, we had to do test run every time. When we started our project, we used to do to test run in university. In our lab, we designed the track on the wooden table according to the specified dimensions on which we had to test again and again to complete our project.

But after some time, it became difficult for all of the students to test on the track at the same time. So, we decided to design the track in our home so that we can try as much as we want in our home.

We deigned the track on the floor with black electric tape. It made easier for us to work easily at our homes.

Other than these platforms we used different platforms for our coding and design that are follows.

- Solidworks
- Tinkercad
- Arduino Software
- Cura

2.5. Assumptions

Following are the assumptions that we did to make our project successful.

- As the system is structured in such a way that there is a potential of mistake in operation, some parameters in the code are hard-coded, resulting in higher efficiency under all scenarios.
- We anticipated a probability of inaccuracy in every testing platform since the IR Sensor results are impacted by sunlight.
- We also assumed that, we could overlook the errors in our results since sensor readings may be skewed by ambient factors.

2.6. Literature Review

[1] This paper describes the design and creation of an autonomous ball-shooting robot for the National Engineering Robotics Contest (NERC) 2008.

The robot can pick up balls from a ball stand and shoot them into goal posts of various heights by navigating through a grid of white reflecting lines. It is equipped with infrared sensors, bump switches, and potentiometers, all of which are interfaced to an AT89C52 microprocessor.

[2] This paper details the creation of an autonomous ball-shooting and maze-navigating robot for the 2012 National Engineering Robotics Competition (NERC). The robot moves through a maze; it chooses the color of the boxes placed at predetermined points around the maze; and it shoot golf balls of the appropriate color in the boxes. A positional Integral-controller has been created using the root locus method to move around the maze without running into the walls. To effectively implementable on a budget-friendly PIC18f4620 microcontroller, the linearization of the Sharp IR sensor is the main foundation upon which the robot's navigation algorithm is built. The robot then makes use of this knowledge in its inventive granular control algorithm to keep itself effectively in the middle of the hallway during its travel. The color of the boxes can be distinguished using a homemade color sensor comprised of discrete parts. According to experimental findings, the robot moves at a pace of 70 cm/s while doing its job without bumping into any walls. These robots are used in industrial pick-and-place activities.

[3] Researchers frequently utilise formal verification, a powerful and cutting-edge technique, to examine software and embedded systems. Recently, academics have employed formal verification to validate robotic systems. This study is based on a robotic fire-fighting system whose design has undergone formal and model checking verification. The robotic system is designed to operate in challenging terrain. There are numerous increasingly sophisticated uses for autonomous robots that can automate our lives. For instance, a robot can help us in situations when human intervention could be harmful or dangerous. This study intends to create a prototype model that will show how a

firefighting robot operates utilising a variety of sensors. This is accomplished by creating an arena with various modules for the robot model to navigate. Because robotic formal verification is a powerful technique that can find and remove bugs from embedded systems in the design stage, robotic systems pertaining to fire-fighting mechanisms have not previously been formally verified, so the model under study leads to innovation in the design and working of autonomous robots.

[4] This study presents a Line Following Robot (LFR) application for a library inventory management system (LIMS). To follow the line path established for book shelf configurations in libraries, a line following robot with sensor operated motors is created. The robot is equipped with a barcode reader that captures barcode information from the vertically stacked books and compares them to the search criteria. If the robot locates the book that needs to be located, it informs the librarian or the person who is visiting the library, where the robot is being used for searching, of where the book is located. The robot halts and emits an alarm if it encounters any obstacles while conducting the search operation. The pre-programmed data in the robot can be used to reorganize misplaced books, keeping the books in order. As a result, the task of organizing the books is aided and made simpler, and the manual regular work performed by library staff is decreased.

[5] In this research, the design of a unique high-speed, low-cost autonomous line- following robot has been presented and put into practice. This robot combines human expert knowledge and experiential data acquired through neural network training. The shortcomings of past designs, such as oscillations in motion, handling of exceptions, high cost, and excessive energy consumption, are eliminated by the proposed autonomous robot. The robot can sense the track ahead and anticipate a turn thanks to a revolutionary square-topology infrared sensor matrix. We describe a neural network-based control method, a learning vector quantization-based strategy, and a reinforcement learning-based strategy for controlling high-speed line following robots.

With just 4 neurons, the final design's hybrid neural network-based control technique smoothly followed the benchmark track. The control system was implemented using an inexpensive 8-bit microcontroller.

[6] The methods for analyzing, designing, controlling, and enhancing the health care are described in this study. System of management. a robot in a line carrying medicine has been created to ensure that the patient receives the drug any time they require it. A line- following robot is a digital technology that is able to recognize and follow the floor- drawn line. The line is typically described as a predetermined path that can be a line that is clearly discernible, such as a black line on a white background color contrast. a resistor sensor that is light-dependent affixed to the robot whose resistance changes according on the light intensity. The robot is equipped with a proximity sensor as well, allowing it to detect obstacles in its path and sound an alarm. The best thing this technology can

offer is the opportunity to contact someone 24/7. This technology concentrated on providing secure, fast, effective, and timely patient centered and equitable.

Chapter 3: Implementation

3.1. Hardware

3.1.1. Working principle

The robot consists of various components that work together to achieve the desired output. Its functionality relies on a system of sensors and actuators. The sensors include IR sensors and an ultrasonic sensor, while the actuators consist of DC and servo motors.

At the core of the project is a line following robot that is programmed to track pre-defined paths made of black lines. The robot utilizes an Arduino mega microcontroller, which interfaces with an array of IR sensors to guide the DC motors.

To comprehend the relationship between the IR sensors, Arduino mega, and DC motors, it is essential to understand their functionality. An IR sensor is an electronic device that emits light to detect objects in its surroundings. It can measure the heat of an object and detect motion. While we cannot see the infrared radiations emitted by objects, an IR sensor can detect them.

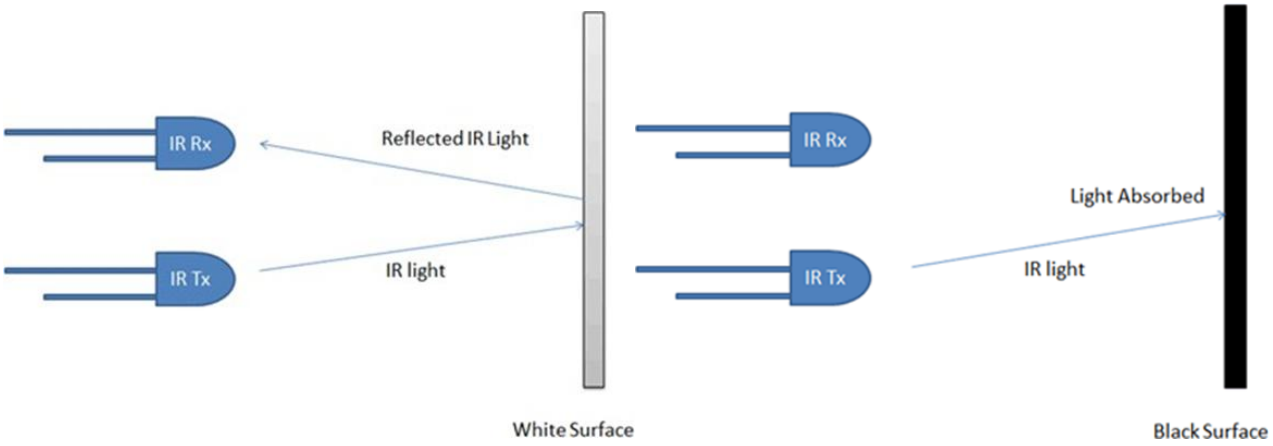
The IR sensor consists of an IR LED (emitter) and an IR photodiode (detector). The photodiode is sensitive to the same wavelength of IR light emitted by the IR LED. When IR light falls on the photodiode, the resistances and output voltages change proportionally.

A typical infrared detection system includes five basic elements: an infrared source (such as an infrared laser or LED of specific wavelength), a transmission medium, optical components, infrared detectors or receivers, and signal processing.

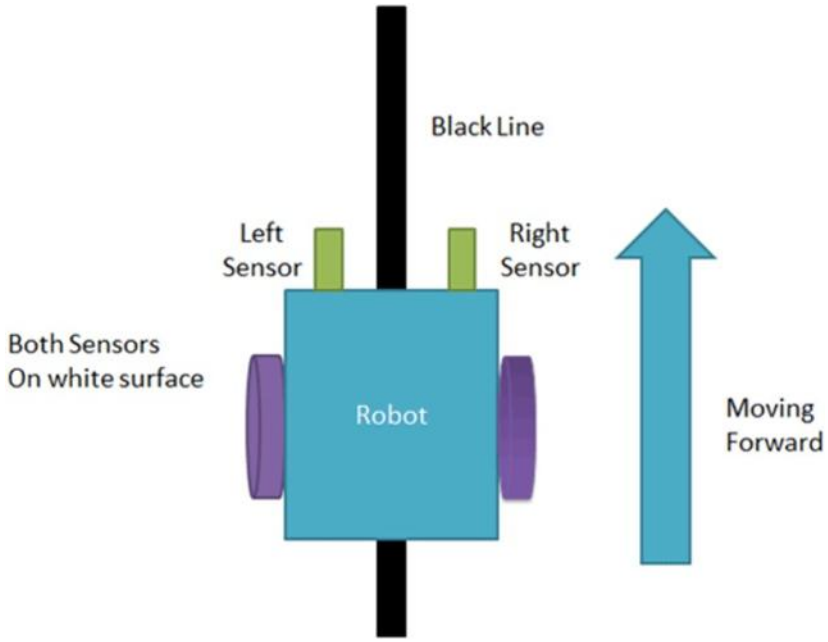
When the IR sensor is operated in digital mode through a microcontroller, it provides an output signal in either HIGH or LOW. When the IR sensor transmits an IR frequency and receives a reflected wave (the magnitude depends on the reflecting surface), it sends a HIGH signal to the Arduino, indicating the reception of an IR signal. However, if the IR sensor transmits an IR wave to a black surface, most of the wave is absorbed, and none is reflected back to the receiver. As a result, the Arduino receives no signal from the IR sensor and outputs a LOW signal.

The robot consists of an array of 3 IR sensors sending digital signals to the Arduino. The LRF is programmed using a system of 3 IR sensors working in conjunction with each other. The IR sensor number 2 (S2) detects and ensures that the robot stays on track. This is done by ensuring that the remaining two sensors stay LOW by staying on the white parts of the track at either side of the black line that the sensor number 2 (S2) follows.

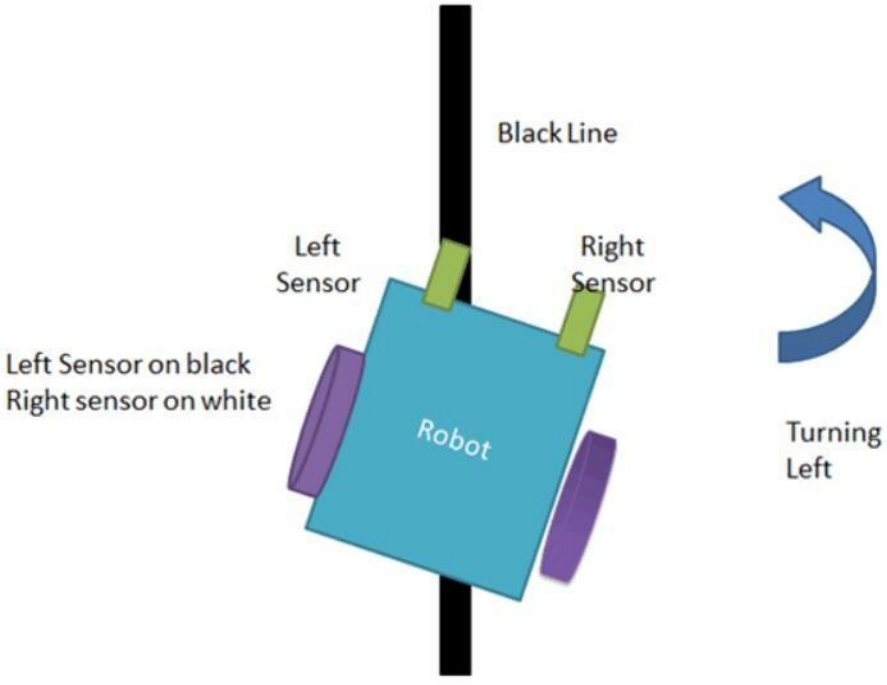
The operation of a line follower is based on the characteristics of light when it interacts with black and white surfaces. When light illuminates a white surface, it is predominantly reflected, whereas a black surface absorbs light entirely. This behavior of light is harnessed in the construction of a line follower robot.



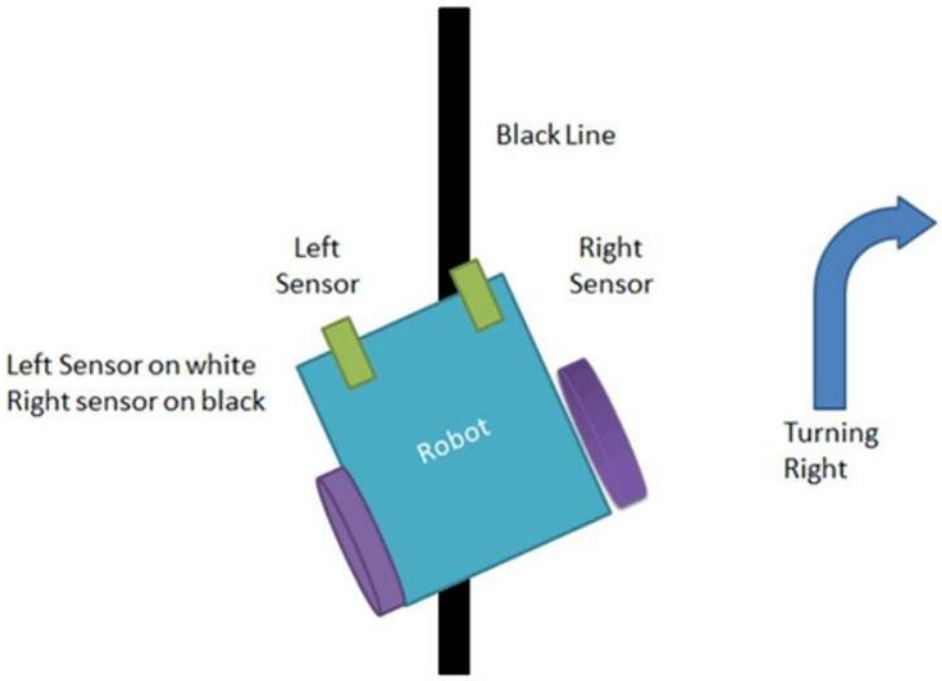
To effectively program the robot, it is vital to grasp the fundamental operations of a basic line following robot. A line follower robot refers to a robot that tracks a specific path by employing a feedback mechanism for control.



This is how an LFR moves forward, by having its left and right sensors ON while the center sensor is OFF on the black line.

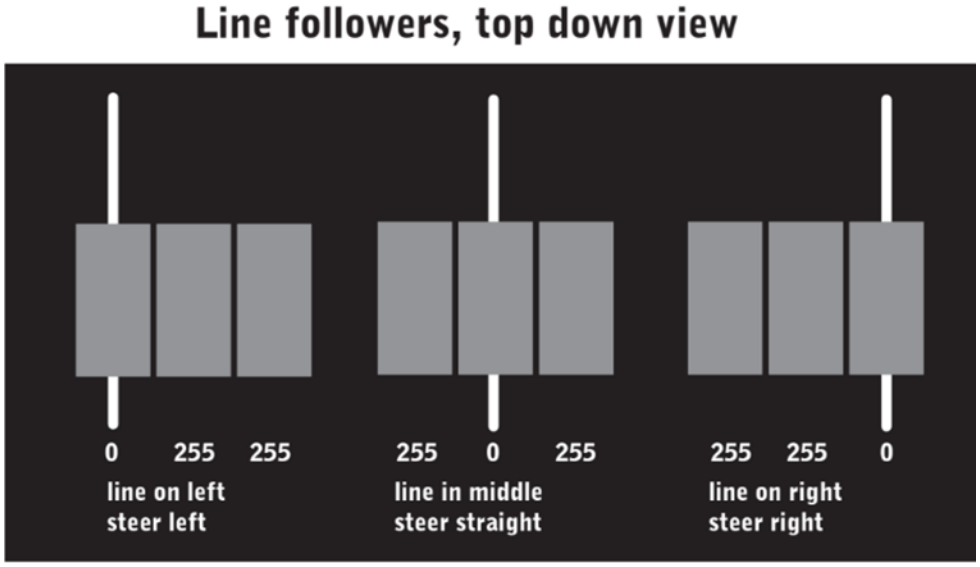


This is how an LFR turns to the left. When the sensors on the left turn off as they fall on the black line, the algorithm dictates that the LFR turns left.



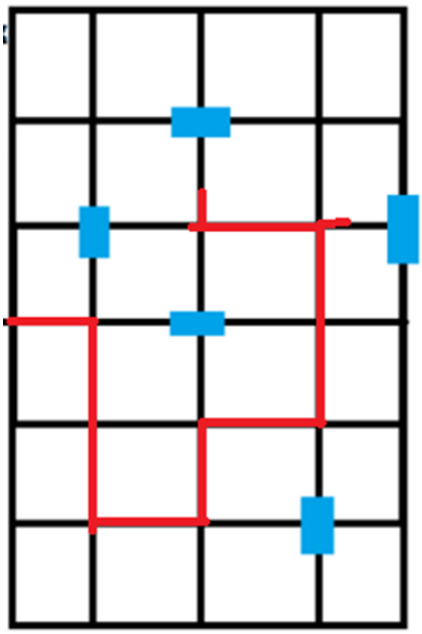
The same happens when turning right, only this time the right sensor goes off and the robot moves right.

These sensor readings are sent to the Arduino which then sends data to the motor driver and according to the algorithm we have made, the LFR moves.



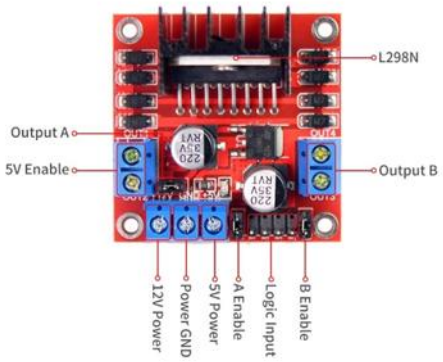
Input		Output				Movement
Left Sensor	Right Sensor	Left Motor		Right Motor		Of Robot
LS	RS	LM1	LM2	RM1	RM2	
0	0	0	0	0	0	Stop
0	1	1	0	0	0	Turn Right
1	0	0	0	1	0	Turn Left
1	1	1	0	1	0	Forward

This table in simple terms shows how a two sensor LFR functions. Since our LFR is a three sensor one, the truth table for it would have 32 outcomes. Our algorithm can be studied in Appendix A, for the sensor values. As the sensors receive values for the path, the motor driver enable pins receives signals from the Arduino which allows it to turn or move forward.



As the illustration above shows, there are 9 total junctions the LFR passes over when traversing the path. These junctions are used as a variable storage system to allow us to hardcode its path. At the first junction, a value of 1 is stored and the LFR moves forward. At the second junction, all 3 sensors turn ON and turns right. On junction 3 LFR moves forward then on junction 4 turn left. On junction 5 robot shoots and turn left. On junction 6 robot shoots and turn right, and so on.

L293D Motor Driver Shield Hardware Overview



L298n Motor Driver Shield Hardware Overview

Popular dual H-bridge motor driver modules like the L298N are frequently used in robotics and electronics projects. It is especially made to easily control stepper motors or DC motors. Two H-bridges make up the module, each of which has two independent motors it can drive both forward and backward.

The L298N module offers a simple and effective method for controlling motors, enabling users to change the speed and direction of the motors in accordance with their application's needs. It is capable of working with a wide range of motor voltages, typically between 5 and 35 volts, and supports a maximum current of 2 amps (or up to 3 amps with the right heatsinking) per channel.

The built-in protection mechanism of the L298N module is one of its key characteristics. It has diodes installed across the motor outputs to help shield the module and other components connected to it from voltage spikes or the back EMF produced by the motors. The module also has onboard voltage regulation, making a variety of microcontrollers and control systems compatible with it.

The L298N is frequently employed in numerous tasks, including robotics, automation, motor control systems, and automotive applications. It offers a dependable and simple method for driving motors, making it a well-liked option among professionals, students, and hobbyists alike.

3.1.1.1. IR Sensor module.

The IR sensor module includes five essential parts like IR Tx, Rx, Operational amplifier, trimmer pot (variable resistor) & output LED. The pin configuration of the IR sensor module is discussed below.

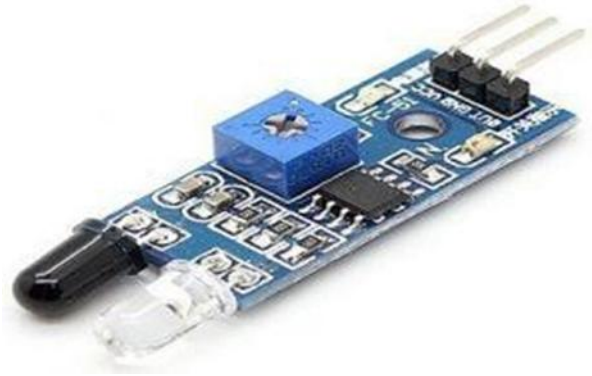


Figure 9

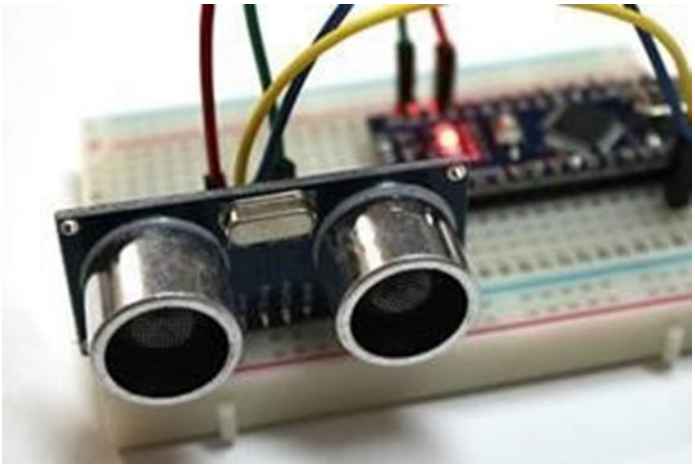
IR Sensor Module

- VCC Pin is power supply input
- GND Pin is power supply ground
- OUT is an active-high o/p

The main **specifications and features of the IR sensor** module include the following.

- The operating voltage is 5VDC
- I/O pins – 3.3V & 5V
- Mounting hole
- The range is up to 20 centimeters
- The supply current is 20mA
- The range of sensing is adjustable
- Fixed ambient light sensor

3.1.1.2. Ultrasonic Sensor



Ultrasonic sensors operate by emitting sound waves at a frequency beyond the range of human hearing. These sound waves are then reflected back, and the sensor calculates the distance based on the time it takes for the reflection to return. This process is similar to how radar measures the time it takes for a radio wave to come back after hitting an object.

There are two types of ultrasonic sensors: those with separate sound emitters and receivers, and those with both functions integrated into a single device. The integrated sensors are smaller in size compared to the ones with separate elements, making them convenient for applications where space is limited.

While radar and ultrasonic sensors can serve similar purposes, sound-based sensors are more easily accessible and affordable, with some available for just a few dollars. In certain scenarios, ultrasonic sensors may detect objects more effectively than radar.

For example, when it comes to transparent materials like clear plastic, radar or light-based sensors struggle to provide accurate readings. In contrast, ultrasonic sensors are unaffected by the color of the object they are sensing and can easily detect clear plastic.

However, if an object is made of a material that absorbs sound or is shaped in a way that redirects the sound waves away from the receiver, the readings from ultrasonic sensors will be unreliable.

If you require precise distance measurements from your sensor, you can calculate it using the following formula:
Distance = ½ T x C
(T = Time and C = the speed of sound)

At 20°C (68°F), the speed of sound is 343 meters/second (1125 feet/second), but this varies depending on temperature and humidity.

3.1.1.3. Dc gear motors.

A gear motor is a combined unit consisting of a motor and gearbox. When a gear head is added to a motor, it reduces the speed while increasing the torque output. The key parameters to consider when dealing with gear motors are speed (measured in rpm), torque (measured in lb-in), and efficiency (expressed as a percentage). To choose the most appropriate gear motor for your specific application, you need to calculate the load, speed, and torque requirements.

ISL Products offers a range of gear motors, including Spur Gear Motors, Planetary Gear Motors, and Worm Gear Motors, catering to various application needs. Our DC motors can be paired with one of our unique gearheads, ensuring a highly efficient gear motor solution.

3.1.1.4. Servo motors.

A servo motor is a precise rotational motor that operates with high accuracy. Typically, this motor comprises a control circuit that provides feedback on the current position of the motor shaft. This feedback enables the servo motor to rotate with exceptional precision. When you need to rotate an object to specific angles or distances, a servo motor is employed. It essentially consists of a basic motor

integrated into a servo mechanism. If the motor is powered by a DC power supply, it is referred to as a DC servo motor. Conversely, if it runs on AC power, it is known as an AC servo motor. In this tutorial, we will focus solely on the working principles of DC servo motors.

Apart from these primary classifications, there exist various other types of servo motors based on different gear arrangements and operating characteristics. Servo motors commonly include a gear setup that allows for high torque capabilities in compact and lightweight packages. These features make them suitable for a wide range of applications, including toy cars, RC helicopters and planes, robotics, and more.

Servo motors are typically rated in kg/cm (kilograms per centimeter). Hobby servo motors often have ratings of 3kg/cm, 6kg/cm, or 12kg/cm. The kg/cm value indicates the weight a servo motor can lift at a specific distance. For instance, a 6kg/cm servo motor should be capable of lifting 6kg when the load is suspended 1cm away from the motor's shaft. The weight-carrying capacity decreases as the distance increases. The position of a servo motor is determined by electrical pulses, and its circuitry is typically positioned alongside the motor.

Servo Motor Working Mechanism

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

3.2. Software

3.2.1. Coding Algorithm

We tried to code our entire project on codes.

- Hard code
- Normal working code

3.2.1.1. Hard Code:

In hard coding, we described all the things and operations of our project. So, that it will run perfectly on the according to tasks perfectly without any errors. In this coding we coded our project about everything that when it had to turn and where to turn and how degrees it had to turn. If it’s going straight then how long it had to run straight. But the major drawback of the hard code is that we had to calibrate every time we test and run it. And if the dimension of the track changes we had to change the code according to it.

Algorithm:

This project was designed to do the multi-tasking. 1st task that it should perform is to follow any path

with IR sensor modules or with hard coding. 2nd task is to detect the basket at any end point. 3rd task is to detect the color

of the basket either by color or by material. And the last task is to pot 2 balls according to the color of the basket. These are the 4 tasks that our robot has to perform.

3.2.1.2. Normal Working Code

In normal coding we just coded all the electronics according to the condition. And when those are met those components had to run the according to it. The basic task that was Line following was depended on IR sensor modules. So, like those other parts had their own described functions. But the major drawback of this was if any part stops working at any time our whole project stops working.

Chapter 4: Results and Discussions

4.1. Results and discussion

Considering all of the operations the robot had to perform, it was decided that a 3 sensor IR array be used for line detection as it provided the most accurate movement of all sensorconfiguration. We used 3 array IR sensor over 5 arrays due to weren't able to operate 5 array it easily and the logic become complicated. The project at its core is an LFR. It is programed to follow a pre-defined track madeby black line. The IR sensor no 2 (S2) detects and ensures that the robot stays on track. TheLRF is programmed using a system of 3 IR sensors working in conjunction with each other.At the junction all 3 sensors turn ON and turns left. On junction 5,6,9,10 (as defined by the code),the robot shoots the balls into its respective target.

4.2. Components Selection Reasoning:

1. Motors gear motors were selected instead of two gear motors and caster or omni wheel. The reasons are as follows:

    - Enhanced maneuverability: With 4 motors, each wheel can be independently controlled, allowing for precise and agile movements. This enables the robot or vehicle to navigate tight spaces, make sharp turns, and execute complex maneuvers with ease.

    - Improved stability: The use of 4 motors provides better balance and stability compared to a 2-motor setup with a caster or omni wheel. By distributing the weight evenly, the robot or vehicle is less prone to tipping over, especially on uneven surfaces or during sudden direction changes.

    - Increased traction and power: Each wheel in a 4-motor configuration contributes to the overall propulsion and traction of the robot. This results in higher torque and improved grip, which is beneficial for traversing challenging terrains or carrying heavy loads.

    - Redundancy and fault tolerance: In a 2-motor setup with a caster or omni wheel, the failure of a single motor or wheel can severely impact the functionality of the robot. However, with 4 motors, even if one motor or wheel fails, the remaining three can still provide partial functionality and allow the robot to continue operating.

    - Versatility and flexibility: Having 4 independent motors enables different modes of operation. For example, the robot can move in any direction, rotate on the spot, or even perform complex movements like strafing. This versatility makes it suitable for a wide range of applications, including robotics competitions, industrial automation, and mobile platforms.

    - Increased load capacity: With 4 motors, the weight distribution is more evenly spread, enabling the robot to carry heavier payloads without compromising its stability. This makes it suitable for

applications that require transportation or lifting of substantial objects.

2. Larger Chassis size of 8x7.5in selected instead of 6x6 in chassis. The reasons are as follows:

- Increased stability: A larger chassis provides a wider base, resulting in improved stability for the robot car. This stability is particularly beneficial when navigating rough terrain or encountering obstacles, as the larger chassis reduces the risk of tipping over or losing balance.

- Enhanced load-bearing capacity: The larger size of the 8x7.5-inch chassis allows for a higher load-bearing capacity compared to the 6x6-inch chassis. This means the robot car can carry heavier payloads or additional equipment without compromising its structural integrity or performance.

- More space for component integration: With a larger chassis, there is more room available for integrating various components such as sensors, actuators, and batteries. This additional space facilitates easier mounting and organization of the components, simplifying the overall design and potentially allowing for future expansions or upgrades.

- Better accessibility for maintenance: The extra space afforded by the 8x7.5-inch chassis makes it easier to access and maintain the internal components. This simplifies troubleshooting, repair, and upgrades, as there is more room to work with and maneuver tools or hands within the chassis.

- Potential for additional features: The increased size of the 8x7.5-inch chassis allows for the potential integration of additional features or functionalities that may not be feasible with a smaller chassis. This opens up possibilities for incorporating advanced sensors, communication modules, or specialized equipment to enhance the capabilities of the robot car.

3. Acrylic sheet was selected for base material instead of other materials such as wood or metal. The reasons are as follows:

- Lightweight: Acrylic sheets are significantly lighter than materials like wood or metal. This reduces the overall weight of the robot car, resulting in improved energy efficiency, faster acceleration, and easier transportation. Additionally, a lighter chassis reduces the strain on other components and allows for greater agility and maneuverability.

- Easy customization: Acrylic sheets are highly versatile and can be easily cut, drilled, and shaped to fit specific design requirements. This enables easy customization of the chassis, allowing for precise placement of components, optimal weight distribution, and efficient use of available space. Customization is especially important for accommodating sensors, actuators, and other

electronics in robotics applications.

- Good electrical insulation: Acrylic sheets are non-conductive and provide excellent electrical insulation properties. This is important for robot cars that house sensitive electronic components. The insulation properties of acrylic sheets help protect the internal electronics from potential short circuits and electrical interference, enhancing the overall reliability and safety of the robot car.

- Corrosion resistance: Unlike metal chassis that may corrode over time, acrylic sheets are resistant to corrosion. This makes them suitable for use in various environmental conditions, including humid or corrosive environments. A corrosion-resistant chassis ensures the longevity of the robot car and reduces maintenance requirements.

- Transparency and visibility: Acrylic sheets have a transparent nature that allows for easy visual inspection of internal components without disassembling the chassis. This is particularly useful during troubleshooting, debugging, or making modifications to the robot car. The transparency also enables better visibility of LEDs, status indicators, or other visual cues that may be integrated into the chassis design.

- Aesthetic appeal: Acrylic sheets offer a sleek and modern appearance, which can enhance the overall aesthetic appeal of the robot car. This is especially important in applications where the robot car is used for demonstrations, exhibitions, or presentations. The transparent or colored options available with acrylic sheets allow for creative and visually appealing chassis designs.

- Cost-effective: Acrylic sheets are generally more cost-effective compared to materials like metal or certain high-quality woods. This affordability makes them a practical choice, especially for hobbyist projects, educational purposes, or situations where budget constraints are a consideration.

- Easy maintenance and repair: Acrylic sheets are relatively easy to clean and maintain. They can be wiped down with common cleaning agents to remove dust or dirt. In the event of damage, acrylic sheets are also easier to repair or replace compared to wood or metal chassis, as they can be cut to the desired shape and size.

4. Two floor shooting mechanism was selected instead of one shooting mechanism with angle titling feature. The reasons are as follows:

- Reduced shooting time: With two shooting mechanisms, the need for frequent reloading is minimized. Each mechanism can be loaded independently, as there is no need to change the angle of the mechanism and then shoot at the target. Hence reduces downtime and improves the overall performance and effectiveness of the shooting system.

- Redundancy and reliability: Having two shooting mechanisms provides redundancy in case one of them fails or malfunctions. If one mechanism encounters a technical issue, the other one can still continue firing, ensuring that the shooting system remains operational. This redundancy enhances the reliability and resilience of the overall setup.

Chapter 5: Conclusion and Recommendation

Conclusion

In order to maintain its intended course, the robot, which is a vehicle system, may sense its path, move, and alter its location in relation to the line. The line-following robot design in this work uses photodiode sensors and can reliably stay on a black line against a white background. The electromechanical robot is 8 by 9.5 in size. Let's assume that the robot can tell when it deviates from the intended path. The autonomous ball shooting line following robot project requires the team to work together, communicate, and acquire knowledge of mechanical, electronic, and integrated systems.

Chapter 6: Appendices

6.1. Coding

```
#include <Servo.h>        //servo motor library
const int servo1Pin = 12;  //servo motor 1 pin no
const int servo2Pin = 13;  //servo motor 2 pin no

//make servo class
Servo servo1;
Servo servo2;

//define ultrasonic 1 pins
#define trig1Pin A3
#define echo1Pin A4

//define ultrasonic 1 pins
#define trig2Pin A5
#define echo2Pin A8

//define ultrasonic1 parameter
long duration1;
int distance1;

//define ultrasonic2 parameter
long duration2;
int distance2;


int ENA = 7;  //enable pin for motor A
int ENB = 2;  //enable pin for motor B
int IN1 = 6;  //control pin 1 for motor A
int IN2 = 5;  //control pin 2 for motor A
int IN3 = 4;  //control pin 1 for motor B
int IN4 = 3;  //control pin 2 for motor B

 int L = A1;
 int C = A2;
 int R = A3;

```

```
int junction = 0;
int sensorL;
int sensorC;
int sensorR;
void setup() {
 // put your setup code here, to run once:
 pinMode(trig1Pin, OUTPUT);  // Sets the trigPin as an Output
 pinMode(echo1Pin, INPUT);   // Sets the echoPin as an Input
 //ultrasonic 2
 pinMode(trig2Pin, OUTPUT);  // Sets the trigPin as an Output
 pinMode(echo2Pin, INPUT);   // Sets the echoPin as an Input
 //servo 1
 servo1.attach(servo1Pin);
 //servo 2
 servo2.attach(servo2Pin);
 pinMode(ENA, OUTPUT);
 pinMode(ENB, OUTPUT);
 pinMode(IN1, OUTPUT);
 pinMode(IN2, OUTPUT);
 pinMode(IN3, OUTPUT);
 pinMode(IN4, OUTPUT);

 pinMode(L, INPUT);
 pinMode(C, INPUT);
 pinMode(R, INPUT);
}

void forward() {
 analogWrite(ENA, 255);
 analogWrite(ENB, 255);
 digitalWrite(IN1, HIGH);
 digitalWrite(IN2, LOW);
 digitalWrite(IN3, HIGH);
 digitalWrite(IN4, LOW);
}

void left() {
 analogWrite(ENA, 255);  //you can adjust the speed of the motors from 0-255
 analogWrite(ENB, 255);  //you can adjust the speed of the motors from 0-255
```

```
75   digitalWrite(IN1, LOW);
76   digitalWrite(IN2, HIGH);
77   digitalWrite(IN3, HIGH);
78   digitalWrite(IN4, LOW);
79 }
80
81 void right() {
82   analogWrite(ENA, 255);  //you can adjust the speed of the motors from 0-255
83   analogWrite(ENB, 255);  //you can adjust the speed of the motors from 0-255
84   digitalWrite(IN1, HIGH);
85   digitalWrite(IN2, LOW);
86   digitalWrite(IN3, LOW);
87   digitalWrite(IN4, HIGH);
88 }
89
90 void stop() {
91   analogWrite(ENA, 0);
92   analogWrite(ENB, 0);
93   digitalWrite(IN1, LOW);
94   digitalWrite(IN2, LOW);
95   digitalWrite(IN3, LOW);
96   digitalWrite(IN4, LOW);
97 }
98
99 void follow() {
100         if (sensorL == 0 && sensorC == 1 && sensorR == 0) {
101           forward();
102         }
103
104         else if (sensorL == 1 && sensorC == 0 && sensorR == 0) {
105           left();
106         }
107
108         else if (sensorL == 1 && sensorC == 1 && sensorR == 0) {
109           left();
110         }
111
112         else if (sensorL == 0 && sensorC == 0 && sensorR == 1) {
113           right();
```

```
114       }
115
116       else if (sensorL == 0 && sensorC == 1 && sensorR == 1) {
117         right();
118       }
119     }
120
121     void IR() {
122       // put your main code here, to run repeatedly:
123
124       sensorL = digitalRead(L);
125       sensorC = digitalRead(C);
126       sensorR = digitalRead(R);
127     }
128
129     void loop() {
130       // put your main code here, to run repeatedly:
131       IR();
132       if (sensorL == 0 && sensorC == 1 && sensorR == 0) {
133         forward();
134       }
135
136       else if (sensorL == 1 && sensorC == 0 && sensorR == 0) {
137         left();
138       }
139
140       else if (sensorL == 1 && sensorC == 1 && sensorR == 0) {
141         left();
142       }
143
144       else if (sensorL == 0 && sensorC == 0 && sensorR == 1) {
145         right();
146       }
147
148       else if (sensorL == 0 && sensorC == 1 && sensorR == 1) {
149         right();
150       } else if (sensorL == 1 && sensorC == 1 && sensorR == 1) {
151         forward();
152         junction++;
```

```
153
154         if ((junction == 1)) {
155           right();
156           delay(800);
157
158         } else if (junction == 2) {
159           forward();
160           delay(400);
161           follow();
162         }
163
164         else if (junction == 3) {
165           left();
166           delay(800);
167         }
168
169         else if (junction == 4) {
170           stop();
171           delay(500);
172           shoot();  //shoot
173           delay(1000);
174           left();
175           delay(800);
176         } else if (junction == 5) {
177           stop();
178           delay(500);
179           shoot();  //shoot
180           delay(1000);
181           right();
182           delay(800);
183         } else if (junction == 6) {
184           left();
185           delay(800);
186         } else if (junction == 7) {
187           forward();
188           delay(800);
189         } else if (junction == 8) {
190           stop();
191           delay(2000);
```

```
192           right();
193           delay(800);
194           delay(500);
195           shoot();  //shoot
196           delay(1000);
197           left();  //uturn
198           delay(800);
199           forward();
200           delay(800);
201         } else if (junction == 9) {
202           right();
203           delay(800);
204           stop();  //shoot
205           delay(500);
206           shoot();  //shoot
207           delay(1000);
208           left();
209           delay(800);
210           stop();  //shoot
211           delay(500);
212           shoot();  //shoot
213           delay(1000);
214           left();
215           delay(800);
216           stop();
217           delay(10000);
218         }
219       }
220     }
221     void shoot() {
222
223       // Clears the trigPin
224       digitalWrite(trig1Pin, LOW);
225       delayMicroseconds(2);
226       // Sets the trigPin on HIGH state for 10 micro seconds
227       digitalWrite(trig1Pin, HIGH);
228       delayMicroseconds(10);
229       digitalWrite(trig1Pin, LOW);
230       // Reads the echoPin, returns the sound wave travel time in
```

```
    microseconds
231         duration1 = pulseIn(echo1Pin, HIGH);
232         // Calculating the distance
233         distance1 = duration1 * 0.034 / 2;
234         // Prints the distance on the Serial Monitor
235         Serial.print("Distance1: ");
236         Serial.println(distance1);
237
238         // Clears the trigPin
239         digitalWrite(trig2Pin, LOW);
240         delayMicroseconds(2);
241         // Sets the trigPin on HIGH state for 10 micro seconds
242         digitalWrite(trig2Pin, HIGH);
243         delayMicroseconds(10);
244         digitalWrite(trig2Pin, LOW);
245         // Reads the echoPin, returns the sound wave travel time in
    microseconds
246         duration2 = pulseIn(echo2Pin, HIGH);
247         //Calculating the distance
248         distance2 = duration2 * 0.034 / 2;
249         // Prints the distance on the Serial Monitor
250         Serial.print("Distance2: ");
251         Serial.println(distance2);
252
253         if ((distance1 < 50 && distance1 > 0) && (distance2 > 60)) {
254           delay(1000);
255           servo1.write(60);
256           delay(200);
257           servo1.write(0);
258           delay(1000);
259         }
260

261         if (((distance2 < 50 && distance2 > 0) && (distance1 < 50 &&
    distance1 > 0)) || ((distance2 < 50 && distance2 > 0) && (distance1 > 60))) {
262           delay(1000);
263           servo2.write(30);
264           delay(150);
265           servo2.write(0);
```

```
266          delay(1000);
267        }
268      }
269
```

6.2. List of Components & Price List

6.2.1. List of Components

There were many components used in making this project. Some of the components were easily available at electronic stores but some of them were veryhard to find at any store. For those components we found much difficulty. So, wehad to visit the stores frequently and many other places so that we could get thosecomponents.
List of electronic components are as follows.

- Arduino At-mega 2560.
- Motor Shield.
- IR Sensor module.
- Ultrasonic Sensors.
- Dc gear motors.
- 2 Servo motors.
- Battery holders
- □□DC motors

List of mechanical components are as follows:

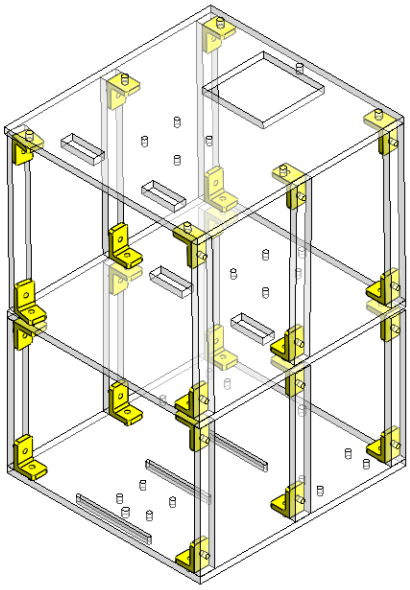
- Screws.
- Nuts.
- Rubber Tires.

6.2.2. Bill of Materials

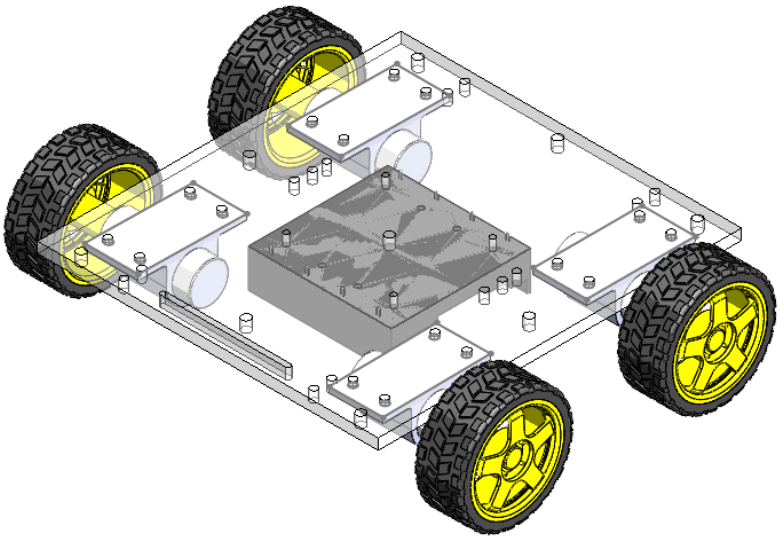
Sr.no	Item	Quantity	Rate	Price
1	Arduino Mega	1	3700	3700
2	Blue Gear Motors	4	450	1800
3	Ultrasonic sensor	2	200	400
4	Servo Motors	2	300	600
5	Acrylic Sheet	2*2ft(6mm)	1500	1500
6	DC Motor	4	200	800
7	3D Printed Parts	Many	10000	10000
8	Battery	3	300	900
9	IR Sensor	1	450	450
Total				20,150

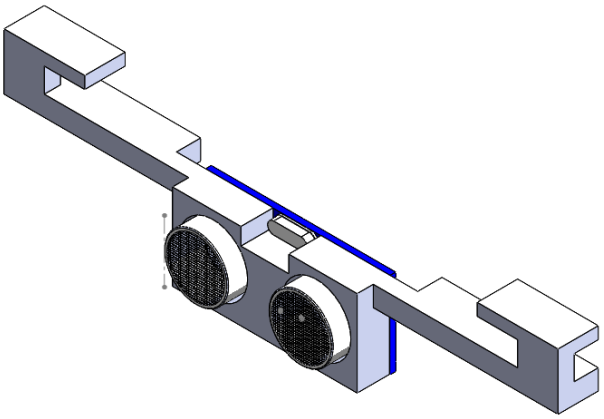
6.3. Mechanical part fabrication layouts

We first designed our base on Solid works. That helped us to narrow downour minds that how we had to design our base in practically. Those designs are as follows.

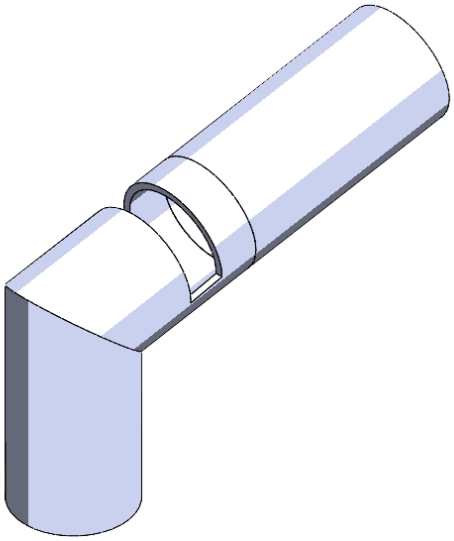


There are 3 floors in the structure, base, first floor and second floor of width of 6mm each. The base contains the Arduino, motor driver, IR sensor and the underneath the base, 4 gear motor are placed along with the battery holder.
The first floor and second floor contain the shooting mechanism. However, the second floor contain additional things such as dual battery holders.
The yellow angle brackets are total 18 in numbers and are made of PLA (3D printed part).

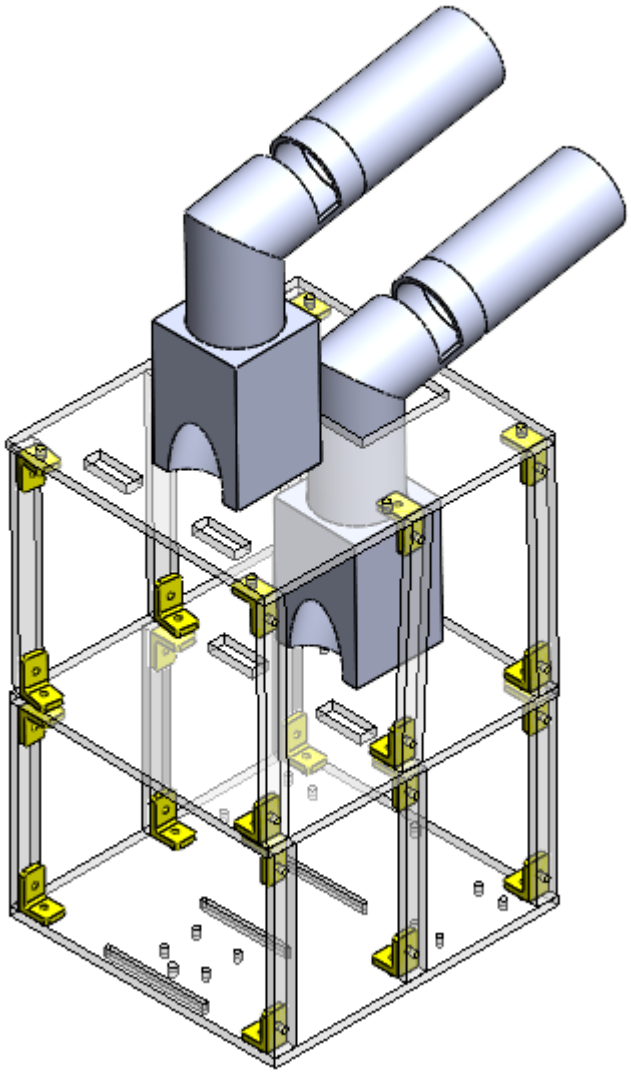




Ultrasonic holder (3D printed)



Ball feeder



structure with shooting mechanism

6.4. Data Sheets

6.4.1. Arduino UNO

Microcontroller	UNO
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega has the capability to be powered either through the USB connection or an external power supply, with the appropriate source being automatically selected. The external power can be provided by an AC-to-DC adapter (commonly known as a wall-wart) or a battery. To connect the adapter, simply insert a 2.1mm center-positive plug into the power jack on the board. For battery power, you can connect the battery leads to the Gnd and Vin pin headers of the POWER connector.

The board is designed to function with an external supply voltage ranging from 6 to 20 volts. However, if the supply voltage is below 7V, the 5V pin may deliver less than the expected five volts, which can lead to unstable performance of the board. Conversely, if the supply voltage exceeds 12V, the voltage regulator may become overheated and potentially cause damage to the board. It is recommended to use a voltage within the range of 7 to 12 volts.

Unlike its previous versions, the Mega2560 utilizes the Atmega8U2 as a USB-to-serial converter instead of the FTDI USB-to-serial driver chip. The power pins on the board serve the following functions:

- VIN: This pin is used to supply input voltage to the Arduino board when an external power source is used, rather than relying on the 5 volts from the USB connection or another regulated power source. You can directly provide voltage through this pin or access it through this pin if utilizing the power jack.
- 5V: This pin provides a regulated power supply to the microcontroller and other components

on the board. The 5V power can be derived from VIN through an on-board regulator or supplied by USB or another regulated 5V source.

- 3.3V: This pin generates a 3.3-volt supply through the on-board regulator. The maximum current draw from this pin should not exceed 50 mA.
- GND: These pins are used as ground connection points.

Memory

The Arduino Mega, equipped with the ATmega2560 microcontroller, possesses specific memory capacities. It has 256 KB of flash memory dedicated to storing code, out of which 8 KB is reserved for the bootloader. Additionally, it has 8 KB of SRAM (Static Random-Access Memory) and 4 KB of EEPROM (Electrically Erasable Programmable Read-Only Memory). The EEPROM can be both read from and written to using the EEPROM library.

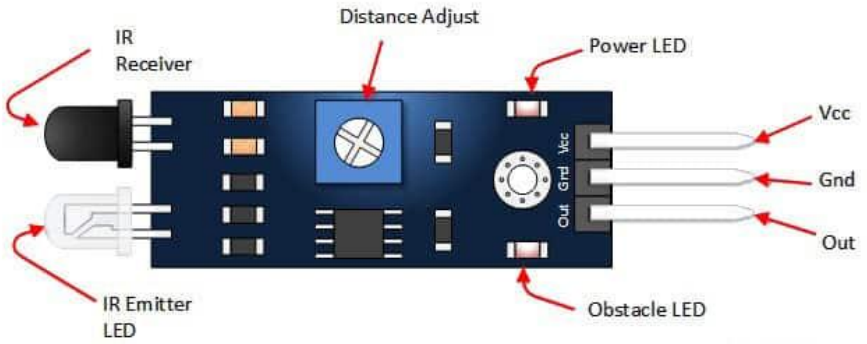
Input and Output

Each of the 54 digital pins available on the Arduino Mega has the capability to operate as an input or an output. This flexibility is made possible by utilizing functions such as pinMode(), digitalWrite(), and digitalRead(). The digital pins operate at a voltage of 5 volts and can source or sink a maximum current of 40 mA. They also come with an internal pull-up resistor, which is initially disconnected, with a resistance value ranging from 20 to 50 kilohms. Additionally, certain pins on the Arduino Mega serve specific purposes:

- Serial: Pins 0 (RX) and 1 (TX); Serial 1: Pins 19 (RX) and 18 (TX); Serial 2: Pins 17 (RX) and 16 (TX); Serial 3: Pins 15 (RX) and 14 (TX). These pins are used for receiving (RX) and transmitting (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: Pins 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt based on a low value, a rising or falling edge, or a change in value. Additional information on configuring interrupts can be found in the attachInterrupt() function documentation.
- PWM (Pulse Width Modulation): Pins 0 to 13. These pins support 8-bit PWM output using the analogWrite() function.
- SPI (Serial Peripheral Interface): Pins 50 (MISO), 51 (MOSI), 52 (SCK), and 53 (SS). These pins facilitate SPI communication and are compatible with the SPI library. The SPI pins are also accessible through the ICSP (In-Circuit Serial Programming) header, which is physically compatible with the Uno, Duemilanove, and Diecimila boards.
- LED: Pin 13. The Arduino Mega comes with a built-in LED that is connected to digital pin 13. Setting the pin to a HIGH value turns on the LED, while setting it to a LOW value turns off the LED.
- I2C (Inter-Integrated Circuit): Pins 20 (SDA) and 21 (SCL). These pins support I2C (also known as TWI, Two-Wire Interface) communication and can be used with the Wire library. It is important to note that these pins are not located in the same positions as the I2C pins on the Duemilanove or Diecimila boards.

6.4.2. IR Sensor Modules

IR Sensor Module Pin Diagram/Pinout



The IR sensor module consists mainly of the 1. IR Transmitter, 2. Photodiode Receiver, 3. LM393 Comparators IC, 4. Variable Resistor (Trim pot), 5. Power LED, 6. output LED.

Pin No	Pin Name	Description
1	VCC	+5 v power supply
2	GND	Ground (-) power supply
3	OUT	Digital Output

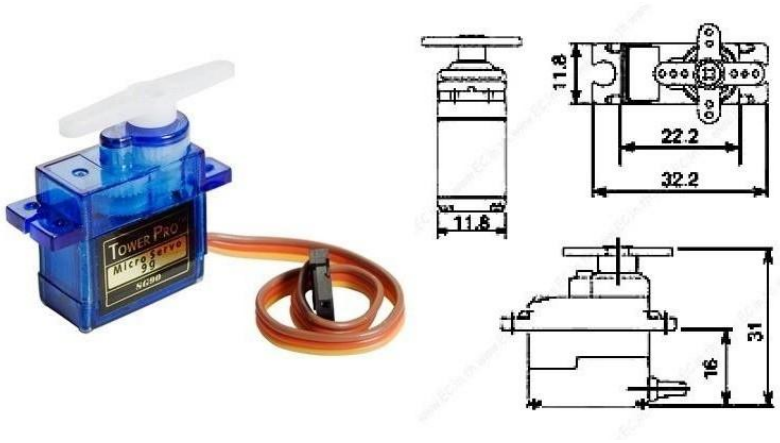
Table 17

6.4.3. DC Gear Motor

Specification	Details
Model Number	DC Gear Motor TT - 120 RPM (Metal Shaft)
RPM (Revolutions Per Minute)	120
Shaft Material	Metal
Gear Type	Spur
Operating Voltage	DC 6V - 12V
No-Load Current	≤ 250mA
Stall Current	≤ 2.0A
Rated Torque	≥ 1.2 kg-cm
Gear Ratio	1:48
Motor Type	Brushed DC Motor
Motor Dimensions	27mm x 20mm x 25mm
Shaft Diameter	4mm

Weight	Approximately 35 grams
Operating Temperature	-10°C to +60°C

6.4.4. Servo Motor



Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 µs
- Temperature range: 0 ºC – 55 ºC

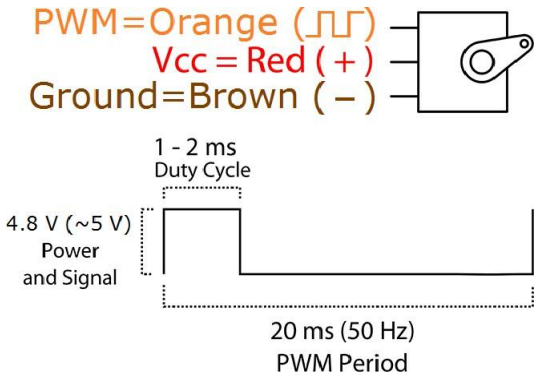


Figure 24

Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

6.5. Calculations

6.5.1. Weight Calculations:

Item	Quantity	Weight(kg)	Joined weight(kg)
Gear Motors	4	0.04	0.16
Tires	4	0.05	0.20
Ultrasonic Sensor	2	0.0085	0.017
Motor Driver	1	0.033	0.033
IR Sensor	1	0.02	0.02
DC Motors	4	0.15	0.60
Acrylic Sheet	-	0.75	0.75
3D Printed Parts	-	0.60	0.60
Total Weight			2.38

6.5.2. Power Calculations:

Component	Operating Voltage	Current Draw	Power Consumption
Arduino Mega	5V	40mA	0.2W
L298N Motor Driver	5V	36mA	0.18W
Ultrasonic Sensors	5V	15mA	0.075W (per sensor)
IR Sensor Array	5V	20mA	0.1W
Servos	5V	100mA	0.5W (per servo)
Gear Motors	3V - 6V	300mA	1.35W (per motor)

Total power consumption: 0.2 + 0.18 + 2(0.075) + 0.1 + 2(0.1) + 4(1.35) = 6.23W.

So according to the calculation the 3 batteries are sufficient for this much power consumption.

And it will run for 5 hours on a single full charge.

7. References

https://ieeexplore.ieee.org/abstract/document/7396402

https://ieeexplore.ieee.org/abstract/document/8967388

https://ieeexplore.ieee.org/abstract/document/5706151

https://www.sciencedirect.com/science/article/abs/pii/S0045790615002190

https://d1wqtxts1xzle7.cloudfront.net/55896243/IJARCET-VOL-2-ISSUE-8-2446- 2450-with-cover-page-v2.pdf?Expires=1655330265&Signature=XLScJyn3qbXx-q2a8UmoLA3NKCuwjRjYwTK~AG4C7xJUJTqL-mBCZIYTjA7wYmLnfi7cZ5buVVdTzrrWkkQOCXqeprvFsPNCJr94nMLSf3b9S7ml82cP8l5G~B1pqc7d19W8HlhE5sKHNnuqOflabdeabTcunK6-efzg04ivtQHU9b6wb~kXKp8FhcYp7~XB1-dQmrR5i-riMH13G-CuAi2jQIyYID38qdlhrqpChoXq4fmJ3EdiCOIPeS9viMCbV3eVp5i5JfWf3oBJ2LVNshOBQdouoi4kyevz83l4XuAoEP- 8SQTphUYZnO3UgsWP4AEBX5h9k1zCXSAVKQgIWA&Key-Pair- Id=APKAJLOHF5GGSLRBV4ZA