ServoBot

Prepared by:

Muhammad Ahmad - 19238

Muhammad Ahmad Uzair - 15086

Dawer Ahmad Shaikh - 14280

Sohail Nasir - 15456

Project Advisor:

Engr. Roomana Yousaf

Final Year Project Report

Department of Electrical Engineering/Biomedical Engineering

Riphah International University, Pakistan

Spring 2023

Riphah International University, Pakistan

Servo Bot

A Thesis Presented to

Riphah International University, Pakistan

In partial fulfillment Of the

requirement for the degree of

B.S. Electrical Engineering

By

Muhammad Ahmad
19238

Muhammad Ahmad Uzair
15086

Dawer Ahmad Shaikh
14280

Sohail Nasir 15456

Spring, 2023

Declaration

We, hereby declare that this project neither as a whole nor as a part thereof has been copied out from any source. It is further declared that we have developed this project and the accompanied report entirely based on our efforts made under the sincere guidance of our supervisor. No portion of the work presented in this report has been submitted in support of any other degree or qualification of this or any other University or Institute of learning, if found we shall stand responsible.

Signature: ______________

Name: Muhammad Ahmad

Signature: ______________

Name: Muhammad Ahmad Uzair

Signature: ______________

Name: Dawer Ahmad Shaikh

Signature: ______________

Name: Sohail Nasir

Riphah International University, Pakistan

Spring 2023

Department of Electrical Engineering

Riphah International University, Pakistan

The project **Servo Bot**, presented by:

1 Muhammad Ahmad	19238
2 Muhammad Ahmad Uzair	15086
3 Dawer Ahmad Shaikh	14280
4 Sohail Nasir	15456

under the supervision of their project advisor and approved by the project examination committee, has been accepted by the Riphah International University, Pakistan, in partial fulfillment of the requirements for the four-year degree of **B.S. Electrical Engineering.**

(Romana Yousaf)

Lecturer, Project Advisor

(Dr. Zahid)

Associate professor & HOD

Department of Electrical Engineering

Dedication

In the chronicles of achievement and perseverance, this work stands as a testament to the power of dedication and determination. With heartfelt reverence, we dedicate this endeavor to a multitude of individuals and ideals that have shaped its journey.

To our families, whose unwavering support and boundless love have been the cornerstone of our pursuit, we extend our deepest gratitude. Your encouragement and belief in us have been the wind beneath our wings, propelling us forward even in the face of challenges.

To our friends, companions in both joy and tribulation, who have stood by us with unwavering camaraderie, we offer our heartfelt appreciation. Your presence has not only lightened the burdens but also amplified the joys along this scholarly expedition.

To the mentors and guides who have shared their wisdom generously, shaping our intellectual horizons and fostering an environment of inquiry, we express our profound respect and admiration. Your guidance has illuminated the path to understanding and knowledge, enriching both the project and our personal growth.

To the pursuit of knowledge and innovation itself, we pledge our continued commitment. This work stands as a humble offering to the ever-evolving realm of human advancement, an endeavor to contribute to the collective tapestry of ideas that propel our society forward.

As we inscribe these words, we recognize that this achievement is not merely the product of isolated effort but the culmination of a network of connections, a symphony of collaboration, and a symposium of inspiration.

May the ripples of our diligence extend far beyond these pages, reaching minds yet unacquainted with the ideas contained herein. May our dedication serve as a spark for others to embark upon their journeys of exploration and creativity.

In reverence to the dreamers, the believers, and the seekers of knowledge, we present this work. May it stand as a beacon of persistence, an ode to cooperation, and a monument to the pursuit of excellence."

Acknowledgment

We owe a great deal of gratitude to the following people for their contributions to the accomplishment of our project thesis, "ServoBot." Their advice, knowledge, and constant support have all been extremely helpful in making this study project a success.

First and foremost, we want to express our sincere gratitude to Romana Yousaf, the engineer who oversaw our project. Her outstanding guidance, insightful advice, and unwavering support have been extremely helpful during this journey. Her direction in determining the project's course, assessment and constructive criticism of the work, and encouragement of original thought have been inspirational.

Also, we want to express our profound gratitude to the instructors, personnel, and resources at RIPHAH INTERNATIONAL UNIVERSITY. Their commitment to creating a space that is conducive to learning and research has immensely helped both our academic and personal growth.

Thank you all for your unwavering support and contributions.

Abstract

Robotics and artificial intelligence are rapidly advancing, which has created new opportunities for the restaurant business to undergo a revolution. To improve overall service effectiveness, customer satisfaction, and smooth operational integration, this thesis investigates the adoption and impact of ServoBot, an autonomous serving robot, in restaurants. Our robot, incorporates the sleek and appealing design elements of the Bella Bot, creating a visually attractive and approachable appearance. It features a compact and lightweight body, outfitted with a range of sensors and intelligent capabilities to navigate in dynamic environments while ensuring the safety of both the robot and the humans around it.

The ServoBot is designed to excel in customer service applications. It possesses an interactive touch-screen display, enabling users to interact with the robot through an intuitive and user-friendly interface. With its sophisticated voice recognition and natural language processing capabilities, the robot can understand and respond to customer inquiries, provide information, and assist with various tasks. Furthermore, the ServoBot integrates autonomous navigation algorithms, enabling it to autonomously navigate through crowded environments, avoid obstacles, and efficiently reach its destinations. It utilizes a combination of computer vision, depth sensing, and mapping technologies to perceive its surroundings, ensuring accurate and secure navigation even in dynamic and unstructured environments.

To enhance its versatility, our robot is equipped with modular attachments and accessories, allowing it to adapt to specific duties and environments. For instance, it can be equipped with trays or shelves for serving food and beverages, or it can be fitted with a medical module for assisting healthcare professionals in delivering medication or gathering vital signs. Overall, the ServoBot represents a significant advancement in human-robot interaction and service robotics. By incorporating the design and features of the Bella Bot, it combines aesthetics, functionality, and intelligence, making it a valuable asset for industries seeking to optimize customer service and operational efficiency. Our research contributes to the field of robotics by presenting a versatile and adaptable robotic system that can revolutionize human-robot interaction across various domains.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -1

1 INTRODUCTION

Robotics has gained notoriety recently as a technology that has the potential to change a variety of sectors, including hospitality, healthcare, retail, and more. The creation of sophisticated robotic systems that can communicate with humans in natural and intelligent ways has emerged as a key topic of study. The Bella Bot, a robot created by Pudu Robotics, is one such prominent example. It has drawn notice for its chic aesthetic and outstanding functionality in customer service applications.

The new robotic system Servo Bot closely resembles the Bella Bot in terms of appearance and operation. It expands on the foundation that the Bella Bot created. We provide practical solutions for industries wanting to boost operational effectiveness and customer service with the aim of further improving human-robot interaction through the use of robotic technology advancements.

The development of robotics and its potential to transform service-oriented businesses are demonstrated by the Servo Bot. We hope to develop a visually arresting and engaging robot that organically draws human contact by embracing the Bella Bot's appealing appearance.

In addition to being aesthetically beautiful, the robot's streamlined and sleek design invites interaction from consumers, bridging the gap between people and technology.

The Servo Bot's attractive form is complemented with a variety of sensors and cognitive features that make it possible for it to deliver flawless customer service. Users may communicate with the robot easily thanks to an interactive touch-screen display, which provides a user-friendly interface that makes task execution and communication easier. The Servo Bot is able to comprehend and reply to consumer enquiries, give pertinent information, and help with a range of activities thanks to superior voice recognition.

Another crucial component of our robot's functioning is autonomous navigation, which is inspired by the Bella Bot's skill at navigating busy spaces. The Servo Bot moves autonomously across dynamic settings by integrating cutting-edge algorithms and sensing technologies, avoiding impediments, and assuring efficient traversal to its targeted destinations.

The Servo Bot uses modular attachments and accessories to increase adaptability. Due to its versatility, the robot can handle various settings and do certain jobs. It can have trays or shelves for serving food and drinks in restaurants, or it can have a medical module to help medical personnel administer medication or gather vital signs. Because of its adaptability, the Servo Bot can adapt and serve a variety of industries.

To sum up, the creation of the Servo Bot marks a significant development in the fields of robotics and human-robot interaction. Our research intends to develop a cutting-edge robotic system that excels in customer service applications by taking design and feature cues from the Bella Bot. The Servo Bot has the potential to disrupt industries by improving consumer experiences and operational efficiency thanks to its aesthetically pleasing design, interactive interface, autonomous navigation capabilities, and modular adaptability.

1.1 Background / Motivation

Numerous industries, including hospitality and customer service, have seen new opportunities arise as a result of the rapid breakthroughs in robots and artificial intelligence. Due to its potential to improve operational effectiveness and client experiences, the use of serving robots in restaurants and other service-oriented organizations has drawn considerable attention.

Successful serving robots that have already had a significant impact on the industry include Bellabot and Pudubot. Bellabot and Pudubot has been accepted by restaurants all over the world thanks to its endearing appearance and interactive capabilities, altering the way food is served and raising customer involvement. These robots drawn notice for its smooth navigation and effective delivery services. Both has become well-known for its contemporary style, interactive features, and capacity to maneuver through crowded spaces, making it an excellent choice for sectors like hospitality, healthcare, and retail. Its success has encouraged scientists and engineers to look more closely at how robotics might improve human-robot interaction and transform service-oriented industries. Both robots can navigate with ease across dynamic surroundings while remaining safe for both themselves and those around them to clever navigation algorithms. It's interactive touch-screen display and sophisticated voice recognition capabilities have shown the possibility of easy-to-use, natural human-robot interfaces. With the use of these characteristics, users may interact with the robot easily, asking questions, getting answers, and asking for help. By proposing a brand-new serving robot called "SERVOBOT" and drawing inspiration from these effective serving robots, we are seeking to advance the fields of human-robot interaction and service robotics.

ServoBot was developed in response to the growing need for sophisticated robotics solutions in the hospitality sector and the rising significance of first-rate customer service. Modern customers anticipate more efficient, dynamic, and engaging customer experiences, which present problems for traditional service methods. By utilizing cutting-edge robotics and research on human-robot interaction, ServoBot seeks to outperform the capabilities of its forerunners. The objective is to develop a serving robot that not only excels in accuracy and efficiency but also builds relationships with people to provide truly great dining experiences. The main goal is to significantly advance the field of human-robot interaction and provide useful advice to businesses looking to enhance their operational efficiency and level of customer satisfaction.

We hope that our research will progress the field of robotics and open the door to new developments in service-oriented companies in the future. Also, we are still interested to see how service robotics develop moving forward, with ServoBot serving as a lasting inspiration and a symbol of innovation. In order to build a future where seamless human robot collaboration introduces in a new era of extraordinary service experiences and shapes the world of tomorrow, it is our hope that this work encourages researchers, engineers, and industry professionals to embrace the potential of robotics and artificial intelligence.

1.2 Problem solving

In response to the threats posed by pandemics like the coronavirus illness and influenza, ServoBot represents a ground-breaking answer. The majority of these highly contagious diseases are caused by contacts between people, necessitating a robotic intervention that lessens direct contact and lowers the risk of transmission.

By utilizing its cutting-edge technological capabilities, ServoBot solves this pressing issue by acting as a reliable tool for disease control and prevention. With its autonomy, ServoBot can complete a variety of activities with the least amount of assistance from humans, lowering the risk of viral spread. In a variety of contexts, including restaurants, ServoBot offers a secure and effective replacement for conventional human interactions while delivering meals or providing other critical services.

ServoBot dramatically reduces human contacts in order to handle the serious issues surrounding pandemics on a worldwide scale. As a result, it significantly contributes to stopping the spread of infectious diseases. Additionally, its incorporation into the restaurant sector offers a beneficial option for rapid and effective food delivery, thereby satisfying the changing needs of contemporary consumers and assuring their safety and wellbeing.

1.3 Scope and limitations.

By avoiding direct human contact and lowering the danger of transmission in a variety of situations, such as healthcare facilities, public areas, and other high-risk regions, ServoBot has a considerable potential for reducing the spread of contagious diseases. The scope of ServoBot includes the foodservice sector, particularly in restaurants, where it provides speedy and reliable food delivery services. It improves operational effectiveness, streamlines the delivery process, and offers clients a practical and dependable option. By offering prompt and precise service, ServoBot seeks to enhance the overall client experience. Its autonomous capabilities and sophisticated navigation systems guarantee effective and precise delivery, resulting in satisfied and devoted customers. The application of ServoBot goes beyond providing medical care and meals. It is applicable to many fields, including hospitality, retail, and logistics, where automation and contact minimization are crucial.

In order to navigate complicated and dynamic surroundings, ServoBot may encounter difficulties. Obstacles, congested areas, or unforeseen events could necessitate human intervention or impact how effectively it operates. The physical capabilities of ServoBot may be constrained when handling jobs that demand manual dexterity or human judgment. It might not be appropriate for complex procedures or jobs that call for human skill. The cost of procurement, programming, and system integration may be a major portion of the initial cost of deploying ServoBot. For best performance, continuous upkeep and technical support are also required. The level of public acceptance and confidence in robotic technologies may determine how widely ServoBot is used. Concerns about data security, privacy, and the human touch in service delivery may make it difficult for it to be accepted and integrated into different businesses.

CHAPTER - 2

2 LITERATURE REVIEW

2.1 Overview of restaurant automation and robotics

Robotics and automation in restaurants relate to the incorporation of robotic systems into many facets of restaurant operations. It entails the employment of robots and automated machines to complete duties usually completed by humans in a foodservice environment. This summary gives a brief glance into the main areas in which automation and robotics have been used in restaurants. Improved software and falling hardware prices, such as those of sensors and circuit boards, have contributed to the expanding use of robotic applications in numerous industries, such as a robot restaurant.

Among the many aspects that contribute to the ideal eating experience, the dining experience offered by restaurants is essential in ensuring visitor happiness. The level of social attraction and engagement, the caliber of the service, and the physical layout of the restaurant can all be rated as the top three factors. The physical surroundings of the restaurant have both a practical and social component, and they have a significant influence on the level of customer service. Customers interact with these forces and create their own meanings and values, which they then communicate through their feelings, thoughts, imagination, and action. The four components of the eating experience include food quality, social & interaction quality, the physical atmosphere of the restaurant, and service quality.

Customer happiness and restaurant reputation are significantly impacted by the quality of the food delivered and the quality of interactions. Customer satisfaction, restaurant reputation, and client engagement are frequently cited as determinants of customer loyalty.

2.2 Existing solutions and related work

Fast-food and quick-service restaurants are increasingly using self-service kiosks, which let consumers place orders and pay for their meals without speaking to a cashier. This improves efficiency, shortens wait times, and streamlines the ordering process. Restaurant inventory management has been improved with the help of various software and hardware solutions. These systems detect expiration dates, keep track of stock levels, and automate order processes using sensors, barcode scanners, and analytical analytics. To keep restaurant premises clean, autonomous robots with cleaning features like floor scrubbing or sanitization capabilities are being deployed. These robots can clean effectively, lessening the workload on staff.

A humanoid robot named Pepper was developed by Softbank Robotics. It is used at many restaurants throughout the world to welcome customers, present menu information, and have basic dialogues with them. To communicate with customers, Pepper uses facial recognition and natural language processing. In some restaurants, autonomous robot waiters have been introduced, capable of bringing orders right to the tables of clients. These robots use sensors and mapping technology to traverse restaurant surroundings, providing precise and effective service.

The most recent version of Pudubot is Pudubot 2. Pudubot 2 uses cutting-edge AI algorithms and advanced robotics technology to give an even more smooth and customized service experience, building on the success of its predecessor. Pudubot 2 is poised to transform the hospitality sector with its modern look and improved navigation features by providing quick and enjoyable client encounters that create a lasting impression. Prepare to be astounded as Pudubot 2 elevates service robots and improves how we connect and eat in service-oriented settings.

2.3 Sensor technologies in restaurant automation

With the use of sensor technology, many operations can be detected, watched over, and controlled in restaurants. The employment of automated or robotic technologies to carry out operations inside a restaurant is known as restaurant automation. These robots can do a wide range of tasks in your company, from serving food to guests to cooking meals. While these robots are in the front of restaurant innovation, it's crucial to keep in mind that they still have some limits.

Infrared or ultrasonic proximity sensors, for example, can identify the presence of things or persons nearby. In crowded restaurant environments, these sensors are utilized to prevent collisions between robots or automated machinery and people. In robotics, contact-free, mid-range distance measurements using ultrasonic (US) and infrared (IR) sensors are often employed in navigation systems for people, mobile robots, and vehicle-related applications. One of the difficult issues with navigation systems is obstacle detection. The performance of the distance measuring sensor changes when barriers made of various materials are present. Ultrasonic sensors assess your proximity to an object using sound waves (echolocation). To assess if an object is there, IR sensors employ infrared light. These sensors' accuracy and dependability are also key differentiators. The vast majority of the time, ultrasonic sensors will provide more accurate and trustworthy data than IR sensors. Mobile robots frequently employ ultrasonic sensors to navigate clear of obstacles.

Food temperatures are monitored and controlled using temperature sensors, such as thermocouples or infrared sensors Thermocouples may be made into a wide range of forms and sizes and are frequently affordable, robust, and lasting. Thermocouples all function in the same way. When subjected to heat, they produce a little voltage. The electrons seek to move about and move away from the heat as the heat is applied to the metal wires in an increasing manner. As the temperature at the detecting point varies, the difference in voltage between the two distinct types of metal wire will rise and fall. To preserve the highest levels of quality and safety, they make sure that food is prepared, stored, and served at the proper temperatures.

load cells measure force or weight. Load cells are used in restaurant automation for duties like portion control, ingredient dispensing, or keeping track of stock levels in storage spaces. A load cell is an electro-mechanical sensor used to gauge force or weight. A wellknown transference between an applied force, material deformation, and the flow of electricity underlies its straightforward yet effective design. They are very adaptable machines that deliver precise and reliable performance in a wide range of applications. It is not shocking that they are now necessary for many industrial and commercial procedures. Effective methods for measuring forces and weights are required by recent developments in robotics, haptics, and medical prosthesis, to mention a few. To suit the demands of this constantly shifting industry, new types of load cells are constantly being developed. Robotics is being advanced to new heights by load cells. High-accuracy and nanoscale applications are now achievable because to new programming languages, software, and materials.

Protection from fire lessens the effects of an unattended fire, protecting people and property. As technology developed quickly throughout the years, new security tools were created. One of today's fastest growing fields of engineering research is robotics. To protect people, the environment, and property, it is in the public interest to find and locate harmful gases. A photoelectric sensor is a component of an optical smoke alarm that tracks changes in an electroluminescent diode's light signal. The signal is continuous and normal under typical circumstances provided there is no smoke. Using a variety of technologies, smoke alarms can sense minute airborne particles to identify fires. They trigger the alarm when they find such particles above a specified threshold. In restaurant kitchens, gas and smoke sensors are essential for safety and monitoring reasons. They sound alarms when dangerous gases or smoke are present, enabling the deployment of safety measures.

2.4 Arduino and its applications in robotics

Hardware and software prototyping were difficult and expensive before the emergence of open hardware platforms like Arduino. These days, a wide range of Arduino boards with various CPUs, sizes, and networking options are available. The hardware for Arduino is now affordable and simple to obtain. The same IDE (Integrated Development Environment), which is available for several OS, is used to program Arduino and is utilized with all boards (Arduino, 2015). This IDE is open source, cost nothing to download, launch, and use. Programming in C/C++ allows users to develop everything from a straightforward software based on procedures in a single file to an intricate object-oriented program spread across numerous files.

The brain of a robotic system is made up of an Arduino board, which directs and coordinates the motion of the actuators, servos, and motors. Users can configure the

Arduino board to receive input from sensors and produce appropriate output signals to operate the robotic components by connecting sensors and actuators to the board. The programming environment for Arduino enables users to specify how robots should behave and make decisions. Robots may work independently or react to specific situations thanks to Arduino, which combines sensor inputs, processing logic, and actuator control. Arduino boards offer an interface for integrating different sensors, such as gyroscopes, temperature sensors, proximity sensors, and ultrasonic sensors. These sensors provide the robot the ability to observe its surroundings, collect data, and decide wisely depending on the information acquired.

DC motors, stepper motors, and servo motors are just a few of the types of motors that Arduino boards may interact with. Arduino permits precise movement and manipulation of robot parts, enabling functions like locomotion, grasping, or rotation. This is done by regulating the speed, direction, and position of motors.

The simplicity and affordability of Arduino make it the perfect platform for robotics prototype and experimentation. Before moving on to more complicated systems, it enables developers to swiftly iterate on robot designs, test various algorithms, and validate concepts. Robotic arms can be precisely moved and manipulated with the help of Arduino boards. Pick-and-place operations and assembly jobs are made possible by Arduino's ability to interface with motors, encoders, and sensors to regulate joint angles, grippers, and other components.

CHAPTER - 3

3 SYSTEM DESIGN & ARCHITECTURE

3.1 System overview and requirements

The Bella Bot by Pudu Robotics served as an inspiration for the technology that this thesis proposes, which is a robotic platform. Our research's goal is to create SERVOBOT, a flexible and intelligent robotic system that can improve human-robot interaction and offer effective solutions in customer service applications. ServoBot features an approachable and aesthetically pleasing appearance together with a small, light body that makes it easy to maneuver in a variety of contexts. In sectors including hospitality, healthcare, and retail, it achieves outstanding performance by utilizing cutting-edge sensors, clever algorithms, and modular attachments.

An interactive touch-screen display, voice recognition, navigation, and modular adaptability are among the Servo Bot's primary features. These capabilities allow the robot to interact with people, move through busy areas, comprehend questions, give appropriate answers, and adapt to various activities and surroundings. ServoBot can precisely sense its environment, avoid obstacles, and travel on its own by combining computer vision, depth sensing, and mapping technology. This ability maximizes operational efficiency while ensuring the safety of the robot and the people it interacts with.

The ServoBot has an interactive touch-screen display to enable intuitive human-robot interaction. A user-friendly interface makes it simple for users to communicate with the robot and ask questions, get answers, and request help. The robot is capable of understanding and effectively responding to user commands and enquiries because to its sophisticated speech recognition. Another important feature of the ServoBot is its modularity. It can be customized to meet the needs of a particular sector by adding various attachments and accessories. For example, the robot can be equipped with trays or shelves to serve food and beverages, or it can have a medical module to help medical professionals with chores like delivering medications or gathering vital signs.

Requirements:

A thorough set of specifications is necessary for the "SERVOBOT" effective development in order to guarantee its performance and functioning. These specifications can be divided into three categories: design, software, and hardware.

Hardware Requirements:

- A portable, easy-to-navigate body that is compact and lightweight.
- Highly developed sensors for sensing and obstacle avoidance, such as proximity and depth sensors.
- A touch-screen display with high resolution for simple human-robot interaction.
- Strong processors and ample memory for decision-making algorithms, natural language processing, and the efficient processing of sensory data.
- A power management system or long-lasting battery to support continuous operation.
- Flexible, adaptable attachment system with modules.

Software Requirements:

- Reliable and effective autonomous navigation algorithms for movement that is both safe and effective in changing surroundings.
- Algorithms for depth sensing and mapping for precise perception and obstacle avoidance.
- A touch-screen interface with an intuitive design.
- A software architecture that is modular to facilitate the fusion of various attachments and accessories.

Design Requirements:

- Design that is aesthetically pleasing and approachable to promote user involvement.
- An ergonomic design that makes things simple to use and maneuver.
- Security measures and procedures to guarantee the safety of the surrounding people as well as the robot.

- Simple upgradability and upkeep for ongoing use.

Figure 1

The "SERVOBOT" can offer a complete solution for industries looking to optimize customer service experiences and operational efficiency by meeting certain hardware, software, and design requirements. By stimulating innovation and creating new opportunities for service-oriented sectors, the suggested system's successful implementation will progress robotics and human-robot interaction.

3.2 Hardware components (ultrasonic, load cells, Arduino, stepper motors, HC-06)

A number of crucial hardware elements are incorporated into the Servo Bot to enable operation and guarantee top performance. The following hardware parts are essential to the robot's functionality:

Ultrasonic Sensor:

The Servo Bot's ultrasonic sensor plays a key part in enabling its ability to avoid obstacles. This high-quality sensor detects and calculates the distance between the robot and potential obstacles in its environment using high-frequency sound waves and echolocation principles. The sensor sends out sound pulses and measures how long it takes for the pulses to return after hitting an object in order to determine the precise distance. The robot gains the ability to sense its environment and proactively avoid impediments in real-time by incorporating the ultrasonic sensor into its sensory system. By enabling the ServoBot to operate effectively and autonomously while avoiding accidents and other possible hazards, this feature improves the safety of the robot.

Load Cells:

The Servo Bot has load cells incorporated within it for precise weight measurement capabilities. The idea of strain gauges and the deformation they undergo under applied force form the foundation of a load cell's operation. Typically, load cells consist of one or more strain gauges, which are electrical resistors that alter their resistance in response to mechanical deformation or strain. The ServoBot can efficiently manage its payload and keep weight restrictions within bounds by adding load cells. This capacity boosts the robot's productivity and enables it to carry out activities requiring precise weight distribution with accuracy and dependability. Load cells are essential for the ServoBot's performance optimization and successful operation in a variety of applications.

Arduino Microcontroller:

The Servo Bot's central processing unit is an Arduino microcontroller. It is in charge of managing numerous hardware elements, carrying out control algorithms, and facilitating communication between the robot's many subsystems. A dependable and adaptable platform for controlling the robot's sensors, actuators, and other ancillary devices is provided by the Arduino microcontroller.

Stepper Motors:

The Servo Bot uses stepper motors to power the movement of the robot's legs or wheels, allowing for precise and controlled locomotion. These motors are ideal for applications that call for precision navigation in changing conditions because they provide accurate position control. The robot can navigate through busy locations while preserving stability and safety thanks to the stepper motors built into the design of the device. ServoBot's main hardware structure is made up of stepper motors, an Arduino microcontroller, and ultrasonic load cells. Together, these parts allow for precision movement, reliable control, and accurate weight measurement, ensuring the robot's success in customer service applications.

The overall hardware configuration of the ServoBot may include additional components such as sensors, actuators, power systems, and communication interfaces, depending on the particular requirements and functionalities of the robot, despite the fact that these hardware components are highlighted as key examples. In order to achieve the goals of the robot and create a trustworthy and effective robotic system, the selection and integration of these hardware components is essential.

HC-06:

A significant advance in the realm of robotics is the incorporation of the HC-06 Bluetooth module to operate the Servobot wirelessly using a mobile smartphone. With the use of this wireless connection feature, users can easily communicate with the Servobot and direct its motions from a distance. Users may send commands to the robot and get real-time feedback on its status and surroundings thanks to the HC-06 module's smooth bidirectional data flow. This ground-breaking approach not only improves user comfort but also broadens the Servobot's uses, allowing it to easily navigate challenging settings and carry out challenging tasks. This innovative integration, which makes use of mobile devices to operate the Servobot, pushes the limits of wireless robotics and paves the way for a more flexible and participatory robotics experience.

Circuit Design:

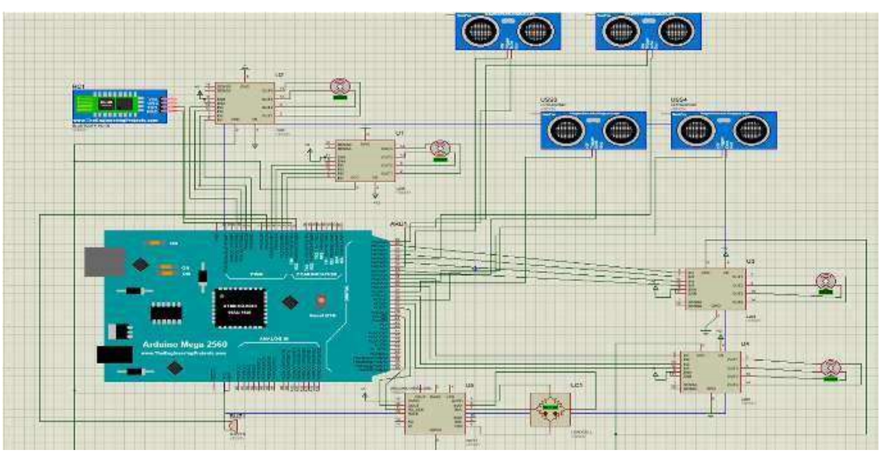

Figure 2

3.3 Software components and algorithms

A variety of software elements and algorithms are incorporated into the Servo Bot to support its intelligent functionality and effective operation. In activities like perception, decision-making, navigation, and human-robot interaction, these software components are essential. The Servo Bot uses the following important software elements and algorithms.

Perception Algorithms:

These algorithms process data from sensors to detect objects and map the surroundings. Distance measuring algorithms enable the robot to autonomously determine the distances to tables, plan its movements accordingly, and execute precise maneuvers. Perception algorithms enable the robot to navigate effectively, avoid obstacles, and interact with its environment in a safe and efficient manner.

Navigation Algorithms:

Advanced navigation algorithms are used by the Servo Bot to enable autonomous mobility. These algorithms compute the best trajectories, avoid collisions, and regulate the robot's movements using information from the perception system. The robot can move efficiently and safely through complex, dynamic settings by utilizing path planning and motion control techniques.

Object Recognition and Manipulation:

The Servo Bot's ability to interact with objects in its environment is made possible via software algorithms for object recognition and manipulation. The robot can pick up, deliver, or arrange goods by using its ability to recognize and categorize objects using sensors. With the help of these algorithms, the robot can adapt to a range of customer service tasks, including providing food and drinks and managing inventories in retail environments.

Human-Robot Interaction (HRI) Algorithms:

HRI algorithms make communication between the Servo Bot and people simple and effective. These algorithms incorporate speech recognition, natural language processing, and dialogue management strategies, allowing the robot to comprehend user instructions and reply to their questions and requests. The robot may participate in meaningful and natural interactions by using HRI algorithms, improving the user experience overall.

Control and Decision-Making:

The Servo Bot's autonomy depends heavily on software elements in charge of control and decision-making. These elements include algorithms for coordinating sensor inputs, controlling actuators in real-time, and carrying out higher-level decision-making procedures. The control and decision-making algorithms enable the robot to make educated decisions and adapt its behavior to dynamic settings by assessing sensor data and environmental inputs.

The software components and algorithms described above should be seen as representative examples; the overall software architecture of the Servo Bot may also include other components unique to the robot's functionalities and goals. The ServoBot can execute difficult tasks, communicate intelligently with humans, and navigate on its own thanks to the integration of certain software components and algorithms.

3.4 Integration and communication between components

The Servo Bot's functionality depends on the various hardware and software components' smooth integration and efficient communication. Communication permits the interchange of information and commands required for coordinated operation, whereas integration guarantees that all components function harmoniously to provide the intended functionality. The Servo Bot's integration and communication features are described as follows:

Hardware Integration:

The ultrasonic sensor, load cells, Arduino microcontroller and stepper motors are just a few of the physical parts that make up the ServoBot. To ensure proper power supply, data transfer, and control signals, this integration entails connecting the hardware components to the appropriate interfaces of the Arduino microcontroller. The hardware components' smooth interaction with one another is made possible by the integration process, which also creates the physical connections.

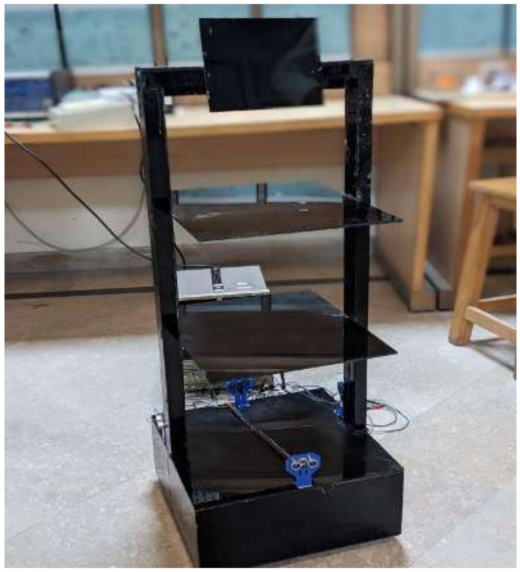

Figure 3

Software Integration:

Software integration entails fusing the ServoBot's different software parts and algorithms into a single, functional system. This integration makes sure that the algorithms for vision, navigation, object recognition, HRI, control, and decision-making operate in unison. Creating a software architecture that allows for data exchange, function calls, and information flow between the various software modules is a step in the software integration process. The robot's functionalities and capacities are coordinated and harnessed to complete the intended tasks by skillfully combining the software components.

Communication Protocols:

Communication protocols are designed to make it easier for the hardware and software components to communicate. The structure, language, and semantics of data exchange and command delivery are specified by these protocols. For instance, the Arduino microcontroller might use well-known communication protocols like I2C or UART to interact with the sensors and actuators. The software components can successfully communicate information, sensor data, and control signals thanks to communication protocols. The establishment of strong and dependable communication protocols ensures that the parts may work together without difficulty and exchange information instantly. The Arduino Mega 2560 serves as the servo bot's core control system, efficiently and precisely monitoring and controlling all attached sensors and motors. On the PCB, there are 4 integrated serial ports. The two pins Rx and Tx are included with each UART serial port. While the Tx is the transmission pin that ensures the transmission of serial data, Rx is the reception pin that ensures the receiving of serial data. There is only one I2C communication protocol on the board. It carries SDL and SCL, two pins. The serial data pin (SDL) and serial clock line (SCL), which provide synchronization of data flow over the I2C bus, respectively, are serial components.

In this revolutionary project, we use the RX and TX pins of the HC-06 Bluetooth module and the UART communication protocol to create a flawless wireless link between the Arduino Mega 2560 and the Bluetooth module. Users are able to remotely operate the ServoBot using their mobile devices thanks to the effective bidirectional data transfer made possible by this. While the Arduino dutifully carries out instructions from the mobile application, the HC-06 module serves as a dependable middleman, permitting the easy interchange of data. With the use of mobile-driven Bluetooth connection, the ServoBot can now be controlled in an immersive, engaging, and varied manner. This innovative approach represents a significant advancement in the field of wireless robotics.

Data Exchange and Synchronization:

Data interchange and synchronization are also involved in component integration and communication. The perception algorithms gather and analyses sensor input, including weight measurements from load cells and obstacle identification from ultrasonic sensors. The higher-level control systems and navigation algorithms are then given access to the processed data in order to help them make decisions. The actuators, such as stepper motors, receive commands and instructions from the control and decision-making algorithms to govern the robot's movement. In order to enable coordinated operation, synchronization mechanisms make sure that data and orders are exchanged accurately and on time.

An essential communication network is established within the ServoBot, which includes an Arduino Mega 2560, stepper motor, load cell, ultrasonic sensor, and HC-06 module, allowing flawless coordination and fine control of its operations. The load cell measures the weight on the ServoBot's tray to keep track of the load capacity while the ultrasonic sensor continually gathers distance measurements to identify any possible obstructions in the ServoBot's route. The Arduino Mega 2560 receives this crucial sensor data via specialized communication protocols UART. The Arduino then uses this incoming data to compute choices, selecting the direction, speed, and step count of the stepper motor, enabling the ServoBot to maneuver deftly around obstacles. Furthermore, the load cell output is constantly checked to avoid going above weight restrictions. By enabling Bluetooth connectivity, the HC-06 module enables users to securely control the ServoBot and get immediate updates. Furthermore, the load cell output is constantly checked to avoid going above weight restrictions. By enabling Bluetooth connectivity, the HC-06 module enables users to securely control the ServoBot and get immediate updates.

Error Handling and Exception Handling:

Error handling and exception handling procedures are included in integration and communication between components. These techniques are intended to identify and address any problems, flaws, or unforeseen circumstances that might occur while the system is in use. Error handling makes sure the system can recover from mistakes, reduce potential hazards, and keep working dependably. Through the use of exception handling, the system is able to react to unforeseen situations and take the necessary steps to protect the stability and safety of the robot.

We ran into a number of difficulties when building this robot, most notably with the obstacle avoidance system. Unintentionally, the motor steps are tallied again when an object is spotted in the robot's safety zone, causing it to stop. Steps are wasted as a result. We have put into place a 'count' variable-based robust approach to deal with this problem. The steps taken when an obstacle is inside the protection zone are added up using this variable as a counter. All of the steps taken during that time period are seamlessly added to the total steps taken before the obstruction was present. This novel method makes sure that motor actions are used as effectively as possible and avoids step waste, thus improving the robot's effectiveness and performance.

We faced major issues with stepper motor movement delays and inaccuracies while implementing the ultrasonic sensor in our project. These problems developed because we had to constantly look for obstructions at each motor step, which caused delays and irregularities in the motor movements. We wisely chose the HCSR-04 library, which has shown to be incredibly suitable for our purposes, to address this pressing issue. The functionality and integration of the ultrasonic sensor were streamlined by the HCSR-04 library, providing a complete and effective solution. We were able to achieve real-time obstacle identification without interfering with the motor movements by utilizing the library's highly efficient features.

Micro stepping techniques were used for the stepper motors to reduce skipped steps, while overcurrent safety devices provided protection from potential harm. Accurate motor motions were made possible through calibrations and maintenance checks. To improve the accuracy of obstacle detection, the ultrasonic sensors' error handling included data filtering and outlier elimination. Procedures for calibrating load cells included compensating for drift and nonlinear responses.

CHAPTER - 4

4 COMPONENT ANALYSIS AND INTEGRATION

4.1 Ultrasonic Sensor principles and functionality:

Ultrasonic sensors are electrical devices that measure distances, identify objects, and offer proximity sensing using high-frequency sound waves (ultrasonic waves). These sensors work on the basis of the reflection and propagation of sound waves.

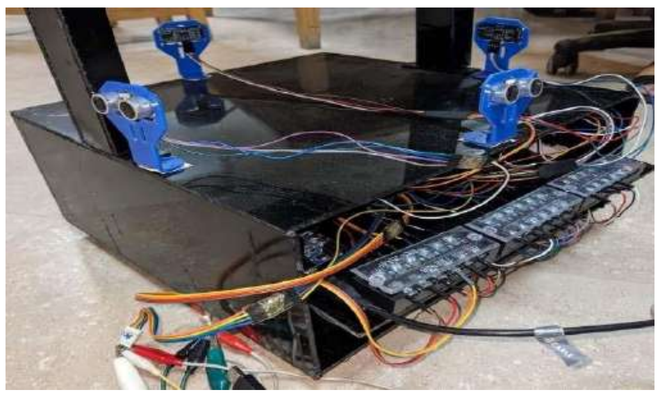

Figure 4

The core principle of how ultrasonic sensors operate is to emit ultrasonic waves from a transmitter and time how long it takes for the waves to travel to an object and return to a receiver. The sensor determines how far apart it is the object based on the time of flight (TOF) of the ultrasonic waves.

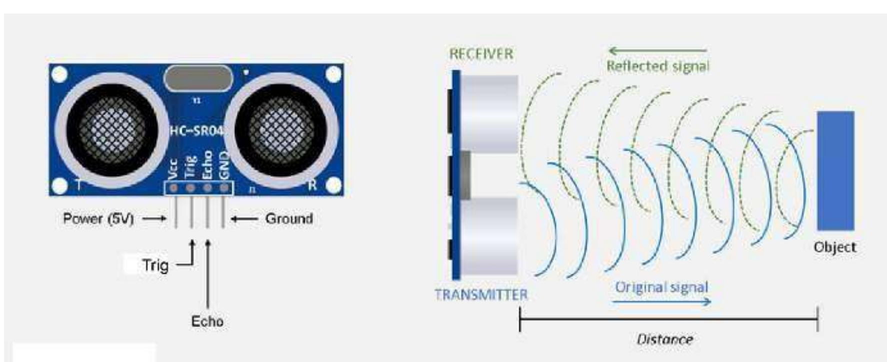

Figure 5

A high-frequency ultrasonic pulse, typically between 40 kHz and several hundred kHz, is produced by the sensor's transmitter. This pulse is aimed towards the intended target or the immediate area. The ultrasonic waves are bounced back toward the sensor when they encounter an item. The reflected waves are captured by the sensor's receiver. The calculated distance is then typically produced, which a microcontroller can read and use for a variety of purposes. An ultrasonic sensor's sensing range establishes the farthest distance at which an object can be consistently detected. Applications like object detection, proximity sensing, obstacle avoidance, level monitoring, and robotics frequently make use of ultrasonic sensors. They have benefits including non-contact operation, a large sensing field, and environmental resistance.

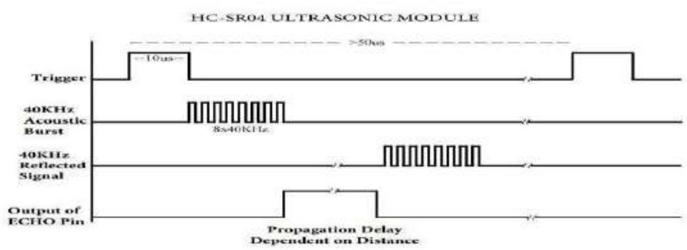

Figure 6

To avoid obstacles, we are employing four ultrasonic sensors in this project. The robot is shielded on all four sides by these sensors. After that, we use this sensor to create a protective border around the robot. As a result, the robot remains around 50 cm (20 inches) away from the obstacle. Ultrasonic sensor is used for obstacle avoidance in this project.

Code:

- void ObstacleDetect () {
- dist1=distanceSensor1.measureDistanceCm(); dist2=distanceSensor2.measureDistanceCm(); dist3=distanceSensor3.measureDistanceCm();
- dist4=distanceSensor4.measureDistanceCm();
- }

4.2 Load Cell principles and functionality:

Using transducers called load cells, mechanical forces or loads can be changed into electrical signals. They are frequently used for measuring and monitoring forces, weights, and loads in many different sectors. The operation of load cells is based on the deformation of a sensing element under applied force and is based on the strain gauge technology. Strain gauges, which are tiny sensors made of thin metallic foils or wire, are frequently used in load cells. The "load-sensitive element" of the load cell, which is a flexible and deformable component, is where these strain gauges are glued to or attached. The load-sensitive element deforms when a force or load is applied to the load cell. The strain gauges are stretched or compressed as a result of this deformation, which alters their electrical resistance.

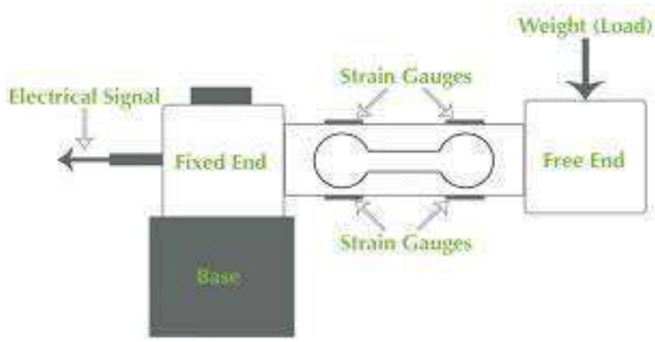

Figure 7

In a load cell, the strain gauges are organized in a Wheatstone bridge pattern. An electrical circuit called the Wheatstone bridge circuit is used to gauge how the resistance of the strain gauges changes over time. The strain gauges' change in resistance when the load is applied results in a modest electrical output signal. In a variety of applications, load cells provide precise and dependable force and weight readings. They are crucial parts of systems that need to monitor, control, or measure exact forces.

For the detection of load on tray of robot load cell used. Load cell not only gives the food occupation bus also limits the load on trays to protect the trays from breaking. Because trays also have limit of bearing weight. If the weight exceeds tray bearing limitations, then there’s a large chance of damage to the trays. Tray can handle upto 1000g at a time. So, load cell serves two advantages detection of occupation of trays also weight measurement of trays.

Code:

- void LoadDetect (){
- static boolean newDataReady = 0;
- const int serialPrintInterval = 0;
- if ((newDataReady)) {
- if (millis () > t + serialPrintInterval) {

- float a = LoadCell_1.getData();
- Serial.print("Load_cell 1 output val: ");
- Serial.print(a);
- if(a>=1000) {
- digitalWrite(alarm, HIGH);
- } else
- {
- digitalWrite(alarm, LOW);
- }
- newDataReady = 0;
- t = millis ();
- }}}

4.3 Calibration and accuracy considerations

To guarantee accurate and dependable distance measurements when utilizing ultrasonic sensors, calibration and accuracy factors are essential. Calibration helps correct for these issues to enhance overall performance. Ultrasonic sensors may experience a variety of conditions that could impair their accuracy. The temperature and humidity of the medium through which the ultrasonic waves travel affect the sound speed. It's crucial to calibrate the sensor for particular operating circumstances in order to obtain reliable measurements. Depending on the surrounding climate, certain ultrasonic sensors may need manual modifications while others may already have built-in temperature adaptation. The actual installation of the sensor and the surrounding area can affect how accurate it is. Ultrasonic waves can be reflected by objects depending on their surface material, shape, and acoustic characteristics. For maximum accuracy, the location of the sensors and the surrounding environment must be carefully considered. The average range of the HC-SR04 sensor is between 2 cm (0.8 inches) and 400 cm (157.5 inches), or 4 meters (13.1 feet). This range may change depending on elements like the target material, the calibration of the sensor, and signal processing.

To establish a connection between the applied force and the output voltage, load cells need to be calibrated. Applying known forces to the load cell during calibration includes recording the associated output voltages. With the aid of this information, a calibration curve or equation is developed that precisely transforms the output voltage into a weight or force measurement. In terms of measuring forces and loads, load cells are renowned for their excellent precision and sensitivity. Depending on the model and design, load cells' range and capacity might range from a few grams to several tons.

4.4 Arduino-based control system

An autonomous robot created to carry out numerous duties is called a ServoBot. An Arduino-based control system is used to achieve precise control and functionality. In order to accurately move and interact with the environment, this control system integrates a load cell with an ultrasonic sensor to operate a stepper motor. The digital and analog input/output (I/O) pins on Arduino microcontrollers can be used to connect to and communicate with other devices, sensors, actuators, and modules. The reading of sensor data and motor control are both possible with these pins. The load cells and ultrasonic sensors provide data to the Arduino microcontroller. To ascertain the weight or force exerted to the trays, it reads the analog or digital signals from the load cells. Similarly, it uses ultrasonic sensors to calculate distance. This data is processed by the microcontroller to obtain pertinent information for regulating the stepper motor. To decide how to operate the stepper motor, the Arduino microcontroller runs a control algorithm that analyzes information from the load cells and ultrasonic sensors. Various capabilities, such as obstacle avoidance based on readings from an ultrasonic sensor or modifying the stepper motor speed and direction depending on load cell measurements, can be implemented by the algorithm. For the ServoBot to run reliably and securely, the Arduino can integrate safety features and errorhandling systems. It can keep an eye on the sensor data and spot unusual or unforeseen events, bringing about the necessary reactions like emergency stops or error alerts.

4.5 Sensor data acquisition and processing

Connect the Arduino's corresponding analog input pins with the load cell. Make that the load cell is grounded and powered according to specifications. If necessary, use an appropriate amplifier or signal conditioning circuit. Using its in-built ADC, the Arduino's analog inputs turn the load cell's analog voltage output into a digital value. To read the voltage value, use the Arduino code's 'analogRead ()' function. In order to measure weight accurately, load cells frequently need to be calibrated. Measure known weights and translate them to the matching analog measurements to calibrate the load cell. Future analog readings into weight values can be obtained using this mapping. After being collected, the load cell reading can then be processed further in accordance with the unique needs of the project. Applying scaling factors, filtering strategies, or any other appropriate mathematical operations may be required to transform the raw data into accurate weight measurements.

Connect the Arduino's digital input/output pins that are appropriate to the ultrasonic sensor. Make that the ground and power connections are made properly in accordance with the sensor's specifications.

Sending an ultrasonic pulse and then timing how long it takes for the pulse to bounce back after striking an item is how ultrasonic sensors operate. Send a trigger signal to the ultrasonic sensor using an Arduino's digital output pin. Then, employ a digital input pin to gauge the interval between the trigger and echo signals. Using the 'pulseIn ()' function in the Arduino code, determine the pulse's (echo signal's) duration. This function delivers the amount of time needed for the pulse to travel to the target and back, allowing for distance calculations. Utilize the airspeed known to be 343 meters per second to convert the pulse duration into the distance. To find the distance to an item, use the proper formula, which is normally two divided by the duration multiplied by the speed of sound.

Depending on the needs of the application, more processing can be done after receiving the distance reading. Extracting the needed information from the sensor data may involve filtering out noise, applying calibration factors, or carrying out additional calculations. In both situations, the sensor data collected can be used to make control decisions, directing the ServoBot to perform particular actions or movements. The microcontroller's control algorithms can process and incorporate these sensor values in real-time thanks to the programming capabilities of the Arduino.

CHAPTER - 5

5 STEPPER MOTOR CONTROL AND MOVEMENT

5.1 Stepper motor principles and operation

Numerous applications that call for precise positioning and control frequently use stepper motors. They work on the basis of step-wise rotation and electromagnetic fields. Multiple electromagnets are arranged in a circular pattern to make up stepper motors. To produce rotational movement, these electromagnets, often referred to as phases or windings, are energized in a particular order. A stepper motor's rotor is made up of a number of poles or teeth that line up with the magnetic field produced by the windings. Stepper motors move in precise steps in response to electrical pulses or impulses. The structure and design of the motor determine the number of steps in each revolution (or every rotation). The quantity and order of pulses given to the windings can be precisely regulated to alter the motor's spinning. A PM stepper motor uses magnetic fields produced by the stator windings to either attract or repel the permanent magnets on the rotor. The rotor rotates in distinct stages as a result of the stator windings being energized in a precise order.

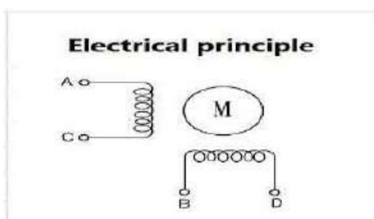

Figure 8

Accurate positioning, speed control, and even bidirectional rotation are attainable by carefully managing the order of electrical pulses given to the stepper motor.

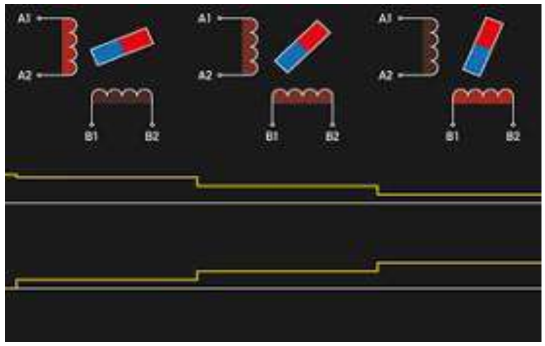

Figure 9

5.2 Control mechanisms and algorithms

A control mechanism and algorithm must be used to control a stepper motor. Connect the Arduino, stepper motor, and an appropriate stepper motor driver. The high-current requirements will be handled by the stepper motor driver, which will also deliver the required signals to control the motor. As a result of its high current capacity and precise control abilities, the L298N module has been selected as the perfect stepper motor driver for our application. We are able to effectively supply the stepper motor with the required power while maintaining dependable and precise control over its motions by utilizing the L298N module. Set the motor's characteristics, including the maximum speed, the number of revolutions per minute, and the numbers for acceleration and deceleration based on the datasheet for the stepper motor. Utilize the L298N module to implement a control algorithm for the stepper motor. The L298N module controls the motor's rotation and direction using a mixture of logic inputs (IN1, IN2, IN3, IN4). Implement the main loop in the Arduino code to control the stepper motor using the algorithm of your choice. This can involve input from the user, sensor readings, or any other elements that affect how the motor moves.

Code:

- Stepper myStepper1(200, 2, 3, 4, 5);
- Stepper myStepper2(200, 8, 9, 10, 11);
- Stepper myStepper3(200,23,25,27,29); Stepper myStepper4(200,31,33,35,37);
- myStepper1.step(1);
- myStepper2.step(1); myStepper3.step(1);
- myStepper4.step(1);

CHAPTER - 6

6 COMPLETE CODE

6.1 Code to control components using arduino

```
//         LIBRARIES        //
#include <Stepper.h>
#include <HCSR04.h>
#include <HX711.h>
#if defined(ESP8266)|| defined(ESP32) || defined(AVR)
#include <EEPROM.h>
#endif
//  Revolutions Defined  //
const int stepsPerRevolution = 200;
const int Table1 = 200 * 4;
const int Table2 = 200 * 7;
const int Table3 = 200 * 7;
const int Table4 = 200 * 7;
const int Table5 = 200 * 7;
const int Table6 = 200 * 7;
//  Distance Measuring Variables  //
float dist1;
float dist2;
float dist3;
float dist4;
int count = 0;
//  Load Cell Variables Declared  //
const int HX711_dout_1 = 38;
const int HX711_sck_1 = 40;
HX711 LoadCell();
const int calVal_eepromAdress_1 = 0;
unsigned long t = 0;
//  UltraSonic Sensor Pin Configuration  //
UltraSonicDistanceSensor distanceSensor1(24, 22);
UltraSonicDistanceSensor distanceSensor2(28, 26);
UltraSonicDistanceSensor distanceSensor3(32, 30);
UltraSonicDistanceSensor distanceSensor4(36, 34);
//  Stepper Motors Pin Configuration //
Stepper myStepper1(200, 2, 3, 4, 5);
Stepper myStepper2(200, 8, 9, 10, 11);
Stepper myStepper3(200, 23, 25, 27, 29);
Stepper myStepper4(200, 31, 33, 35, 37);
```

```
//  Setup To Initialize Components  //
void setup() {
    int alarm = 6;
    pinMode(alarm, OUTPUT);
    myStepper1.setSpeed(150);
    myStepper2.setSpeed(150);
    myStepper3.setSpeed(150);
    myStepper4.setSpeed(150);
    Serial.begin(9600);
    float calibrationValue_1;
    calibrationValue_1 = 733.0;
    #if defined(ESP8266) || defined(ESP32)
    EEPROM.begin(512);
    #endif
    EEPROM.get(calVal_eepromAdress_1, calibrationValue_1);
    LoadCell.begin(HX711_dout_1, HX711_sck_1);
    unsigned long stabilizingtime = 2000;
    boolean _tare = true;
    byte loadcell_1_rdy = 0;
    while ((loadcell_1_rdy + loadcell_2_rdy) < 2) {
        if (!loadcell_1_rdy)
            loadcell_1_rdy =
LoadCell_1.startMultiple(stabilizingtime, _tare);
    }
    if (LoadCell_1.getTareTimeoutFlag()) {
        Serial.println("Timeout, check MCU>HX711 no.1 wiring and
pin designations");
    }
    LoadCell_1.setCalFactor(calibrationValue_1);
    Serial.println("Startup is complete");
}
//  Loop To Perform Logics  //
void loop() {
    if (Serial.available > 0) {
        int data = Serial.read();
        if (data == 1) {
            table1();
        }
        else if (data == 2) {
            table2();
        }
        else if (data == 3) {
            table3();
        }
```

```
        else if (data == 4) {
            table4();
        }
        else if (data == 5) {
            table5();
        }
        else if (data == 6) {
            table6();
        }
    }
}
//   Declaration of Functions to control   //
//  Stepper motor during obstacle and  //
//  Load Cell Measurement   //
//   Distance Measurement from Obstacle  //
void ObstacleDetect (){
    dist1 = distanceSensor1.measureDistanceCm();
    dist2 = distanceSensor2.measureDistanceCm();
    dist3 = distanceSensor3.measureDistanceCm();
    dist4 = distanceSensor4.measureDistanceCm();
}
//   Weight Measurement for  //
//    Protection of Trays    //
void LoadDetect(){
    static boolean newDataReady = 0;
    const int serialPrintInterval = 0;
    if (newDataReady) {
        if (millis() > t + serialPrintInterval) {
            float a = LoadCell_1.getData();
            Serial.print("Load_cell 1 output val: ");
            Serial.print(a);
            if (a >= 1000) {
                digitalWrite(alarm, HIGH);
            } else {
                digitalWrite(alarm, LOW);
            }
            newDataReady = 0;
            t = millis();
        }
    }
}
//   Tables Declaration     //
void table1(){
    for (int i = 0; i <= Table1; i++) {
```

```
        LoadDetect();
        ObstacleDetect();
        if (dist1 < 50 || dist2 < 50 || dist3 < 50 || dist4 <
50) {
            myStepper1.step(0);
            myStepper2.step(0);
            myStepper3.step(0);
            myStepper4.step(0);
            count += 1;
        } else {
            myStepper1.step(1);
            myStepper2.step(1);
            myStepper3.step(1);
            myStepper4.step(1);
        }
    }
    Table1 = count + Table1;
}

void table2(){
    for (int i = 0; i <= Table2; i++) {
        LoadDetect();
        ObstacleDetect();
        if (dist1 < 20 || dist2 < 20 || dist3 < 20 || dist4 <
20) {
            myStepper1.step(0);
            myStepper2.step(0);
            myStepper3.step(0);
            myStepper4.step(0);
            count += 1;
        } else {
            myStepper1.step(1);
            myStepper2.step(1);
            myStepper3.step(1);
            myStepper4.step(1);
        }
    }
    Table2 = count + Table2;
}

void table3(){
    for (int i = 0; i <= Table3
```

6.2 Procedure for executing codes

First, libraries are included into the code, which has several advantages. Libraries make programming easier and increase the functionality of Arduino programs. We have prewritten code blocks and functions available to us thanks to libraries. To start, we utilize a stepper motor library, which allows us to quickly control the stepper motor by just setting its pins. Benefits for ultrasonic echo and trigger are provided by the HCSR04.h ultrasonic sensor library. It chooses for itself the length of the sound wave pulse and the interval between pulses. HX711.h library is used to communicate with hx711 amplifier to read data from load cell. The calibrating factor is stored in an EEPROM library, which is then utilized to measure the right load under varying conditions.

Revolutions are categorized based on the distance between the robot station and the tables. By measuring the distance after properly computing the steps for one revolution, we can define the revolutions of all tables. We define the revolutions by using one revolution steps since a Nema 17 Stepper motor completes one revolution in 200 steps.

Declared variables for output from load cells and distance measurement. Four variables are declared for each of the four ultrasonic sensors, and for each load cell, one output and a clock pin are designated for load measurement. And pin configuration of the ultrasonic sensor for all ultrasonic sensors is declared for Arduino. After the ultrasonic sensor, the stepper motor pins defined on Arduino control the stepper motor.

Setup begins with the definition of the alarm pin and the stepper motor speed. Following this load cell calibration code is necessary because when a load cell is initially turned on, it must be calibrated. Through this calibration, we obtain the calibration factor, which is then permanently stored in the EEPROM and used as a reference each time. Additionally, a load cell outputs becomes correct for all the variations in load. After the setup loop is established, the desired robot's appropriate function is carried out throughout this loop. Following the loop, all tables, load detection, and obstacle avoidance functions are defined. The load detection and obstacle avoidance functions are constructed to be executed in each table function by logic.

CHAPTER - 7

7 DISCUSSION AND CONCLUSION

7.1 Summary of the achieved objectives

The following goals were successfully completed by the ServoBot project. The project's goal of developing a self-driven robot was accomplished. The ServoBot displayed autonomous navigation abilities by fusing cutting-edge sensors, exact motor control, and clever algorithms. Without the aid of outside guidance or manual control, it was able to successfully navigate in a variety of surroundings, design the best routes, and accurately detect impediments. The project's replacement of the standard line-following robot with a more flexible and adaptive solution represented another important goal. The Servobot successfully achieved this goal by combining sensors and algorithms that allowed it to function without the need for pre-established routes or line marks. With no restrictions imposed by predefined pathways, the ServoBot was able to carry out a variety of activities, including table-to-table transfers. Creating a budget-friendly solution was a key goal of the ServoBot project. The project team achieved a budget-friendly design without sacrificing the robot's performance or functionality by carefully choosing cost-effective components and making use of the Arduino microcontroller's capabilities. A wider number of applications and users were made possible by the ServoBot technology, which proved to be a more affordable substitute for conventional line-following robots.

7.2 Contributions and significance of the project

The robotics and automation fields have benefited greatly from the ServoBot project. The ServoBot project developed a self-driven robot with independent navigation, which has helped progress autonomous robotics. The ServoBot exhibited the capacity to function autonomously in dynamic surroundings by merging powerful sensors, clever algorithms, and accurate motor control. With this contribution, there are now more opportunities for the use of robots across a range of sectors, including manufacturing, logistics, and healthcare. This project's primary audiences are the restaurant and medical businesses, and it specifically addresses their unique demands and requirements. The ServoBot can bring food to patrons' tables by independently navigating the restaurant's floor, which eliminates the need for wait staff and improves the overall eating experience.

7.3 Limitations and future improvements

While the ServoBot using load cells and ultrasonic sensors has many benefits, it also has several drawbacks that can be resolved with further development. Here are several restrictions and potential improvement areas:

- The range of ultrasonic sensors for detecting obstacles is constrained. The range may become a problem in bigger contexts, possibly resulting in poor detection or slow reaction. Alternative sensing methods with greater detection ranges, like LiDAR or infrared sensors, can be investigated and implemented into the ServoBot system to get over this restriction.
- Environmental factors like acoustic reflections, background noise, and different surface materials might have an impact on ultrasonic sensors. These elements could affect how precise and reliable obstacle detection is. Future developments might concentrate on adopting sophisticated signal processing methods or applying sensor fusion strategies to reduce the effects of external interference and increase the robustness of the system.
- Although load cells offer precise weight measurements, their accuracy may be impacted by variables including temperature variations, mechanical vibrations, or drift over time. Algorithms for temperature adjustment and calibration can be used to reduce measurement errors and preserve accurate weight readings to increase the precision of weight sensing.
- The ServoBot system's ultrasonic sensors and load cells have certain functions, but adding more sensors can make it even more functional. The performance of the system can be increased by, for instance, integrating vision-based sensors for object recognition or depth-sensing cameras for more accurate navigation and obstacle avoidance.
- Future developments could concentrate on creating sophisticated control algorithms for the ServoBot. To improve navigation effectiveness, obstacle avoidance, and system performance generally, this can include machine learning-

based methods, adaptive control strategies, or algorithms for improved path planning.

The ServoBot is not a general solution that can be used in all restaurants; rather, it is designed to service a particular region. Each restaurant receives a special software preparation that is tailored to its particular requirements.

7.4 Conclusion

In conclusion, the ServoBot project was successful in constructing a self-driven robot that can take the place of older line-following robots. The ServoBot successfully exhibited consistent performance in autonomous navigation, precise movement control, and accurate weight measuring by integrating ultrasonic sensors for obstacle avoidance and load cells for weight sensing. The project makes advances in autonomous robotics, offers a budgetfriendly design, and has the potential to improve processes across a variety of industries.

The ServoBot's capacity to move between tables and navigate autonomously creates prospects for automation and efficiency growth in industries like restaurants and hospitals. It is crucial to be aware of the drawbacks of depending solely on load cells and ultrasonic sensors, such as their limitations in complicated surroundings and the requirement for close contact with objects for precise measurements. These restrictions offer priceless information for upcoming advancements and enhancements.

The incorporation of additional sensors, such as vision systems or LIDAR, to improve navigation and obstacle detection skills may be one of the ServoBot project's future improvements. Through the use of calibration techniques or the investigation of alternative sensor technologies, weight sensing can be made more reliable and accurate.

The ServoBot's movement control and decision-making processes could also be improved by using sophisticated control algorithms and machine learning techniques.

Overall, the ServoBot project provides a practical and affordable solution for a variety of industries while laying the groundwork for future developments in autonomous robots. The project's results advance the field of intelligent automation and open new possibilities for the use of sensors, control systems, and algorithms to improve robot functionality and usability in the future.

CHAPTER - 8

8 REFERENCES

[1] Z. Lin and F. Lin, "Robot-assisted services in hospitality: A review of current status and future prospects," in International Journal of Hospitality Management, vol. 87, p. 102525, 2020.

[2] X. Zhang and J. Su, "Autonomous mobile robots for service tasks in indoor environments: A review," in IEEE Transactions on Automation Science and Engineering, vol. 17, no. 2, pp. 405-420, 2019.

[3] J. H. Yoon and M. J. Kim, "Design and implementation of a service robot platform for intelligent service environments," in Journal of Intelligent & Robotic Systems, vol. 93, no. 1-2, pp. 241-259, 2019.

[4] X. Chen, X. Ren, and H. Zhang, "Autonomous navigation and obstacle avoidance of mobile robots in indoor environments: A survey," in Sensors, vol. 20, no. 12, pp. 3551-3574, 2020.

[5] H. Ahn and Y. S. Kim, "A review on service robots in healthcare and hospitality," in Journal of Intelligent & Robotic Systems, vol. 101, no. 1, pp. 1-20, 2021.

[6] T. C. Huang and R. T. Rust, "Artificial intelligence in service," in Journal of Service Research, vol. 21, no. 2, pp. 155-172, 2018.

[7] M. Li, J. Lin, and X. Li, "Design and implementation of a service robot based on ROS," in Journal of Robotics, vol. 2021, 2021.

[8] F. Wu, Y. Zhang, W. Wang, and H. Ma, "Development and design of a service robot for indoor environments," in Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), pp. 7171-7177, 2019.

[9] Z. Chen and Y. Su, "A novel mobile service robot platform design and its control system development," in Robotics and Autonomous Systems, vol. 125, p. 103365, 2020.

[10] S. Park and D. Jung, "Design and implementation of a social robot for smart environments," in Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 9, pp. 3597-3612, 2019.

[11] A. Bonarini, M. Matteucci, R. Rasconi, and I. Matteucci, "From traditional robotics to soft robotics: A workshop for high school students," in Robot Intelligence Technology and Applications 4, pp. 93-102, Springer, Singapore, 2019.

[12] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," in International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1-13, 1975.

Appendix A
GANTT CHART

ACTIVITY	Oct 2022		Nov 2022	Jan 2023	March 2023	May 2023	June 2023	
Collection of Literature	15th							
Study of Literature		30th						
Analysis of Proposed Scheme			31st					
Preparation of Schemes / Model				28th				
Implementation of Schemes/Model					31st			
Analysis & Simulation						31st		
Result Formulation							15th	
Final Write-up & Thesis Submission								31st

Appendix C
Complex Engineering Problems

Work Package	Characteristic	Features	Score	Compliance (Please add remarks and mark for related preamble)
WP1 *	Depth of Knowledge required	Resolved with **forefront indepth** engineering knowledge (WK3, WK4, WK5, WK6 or WK8) which allows a fundamentals-based, first principles analytical approach.	**3/3**	To gain the basic knowledge of sensor and other components used in the servo Bot, rigorous research was concluded about each item's details, functions and coding during the first phase of our research project. IoT technology was used to observe real time data monitoring including motor status, speed monitoring. The aim was to ensure balancing of bot to final destination with food.

WP2	Range of conflicting requirements	Involve **wide-ranging or conflicting** technical, engineering and other issues.	**1/1**	No conflicting issues involved in the project.
WP3	Depth of analysis required	Have **no obvious solution** and require abstract thinking, originality in analysis to formulate suitable models.	**1/1**	The servo bot has unique features. The circuitry is embedded inside the bottom of servo bot. The components and servo bot performance was checked and tested using testing devices in the lab such as function motors used for testing sensors and other equipment. The project will be commercialized as companies have already shown in interest. Mass manufacturing of the servo bot would require vast space, manpower and more equipment.
WP4	Familiarity of issues	Involve **infrequently encountered** issues	**1/1**	We have used new technology, not used by others in the making of smart servo bot. Coding issues occurred in the walking of servo bot but was resolved.
WP5	Extent of applicable codes	**Beyond codes** of practice	**1/1**	Coding was involved in enabling all three features of the servo bot. Arduino Coding Language was used. All features are functioning properly.
WP6	Extent of stakeholder involvement and level of conflicting requirements	Involve **diverse groups of stakeholders** with widely **varying needs**.	**1/1**	Stakeholders include 80%, restaurants who use robots for work (serving). Unilever Pakistan workers, Food Chains staff, MNCs and hospitals are the major stakeholders.
				The servo bot can be made based on customised requirement.
WP7	Interdependence	Are **high level problems** including **many component** parts or sub-problems?	**1/1**	The project is designed using layered approach, hence no interdependence.
EP1	Consequences	Have **significant consequences** in a range of contexts.	**1/1**	In case of accidents, the servobot automatically stops and do not take further steps if any obstacle comes infront.

EP2	Judgement	Require judgement in **decision** making	**1/1**	Due to sleek designed approach each feature responses as per requirement for decision making.
Aggregate: The servo bot designed for the serving purposes in restaurants and hospitals was successfully made. Its unique features functions properly. The circuits for each feature were assembled independently and embedded inside the servo bot. The project was selected by ignite and nominated for funding. It was also awarded grant by PEC in the ceremony at PEC Lahore.			**11/11 100%**	The components were tested thoroughly ensuring safety measures of the group member in the R&D lab.

Complex Engineering Activities

Preamble	**Complex activities** mean (engineering) activities or projects that have **some or all** of the following characteristics listed below	Score	Compliance (Please add remarks and marks for against relevant preamble)
Range of resources	**Diverse resources** (people, money, equipment, materials, information and technologies). EA1	**1/1**	**Pakistan Engineering Council**
Level of interaction	Require resolution of significant problems arising from interactions between **wide ranging** or **conflicting** technical, engineering or other issues.EA2	**1/1**	Servo Bot uses new technology and there is no conflict.
Innovation	Involve creative use of engineering principles and **research-based** knowledge in **novel** ways. EA3	**1/1**	Servo Bot uses IoT based state of the art components which ensures functionality of its key features.

Consequences to society and the environment (* UN SDGs)	Have **significant consequences** in a **range of contexts**, characterised by **difficulty** of prediction and mitigation.EA4	**1/1**	It can add value to restaurants and hospitals industry.
Familiarity	Can extend **beyond previous** experiences by applying **principles-based** approaches.EA5	**1/1**	All the functions are easy to use.
Aggregate: 1. The helmet is Unique, as it is smart and can send real time information alerts to international and local recipients in case of over speeding, and bike crash. 2. It identifies whether rider is wearing a helmet or not. It can send location, time and date of the incident to the nearest contacts.		**5/5 100%**	Servo Bot ensues it add value to restaurants and hospitals industry.

Comments:

80% of the restaurants in Pakistan can use the servo Bot to increase their customer service also hospitals can use it to deliver medicines etc. We therefore need to normalize this by inculcating a culture of robots so that not just only locally, it will also have an impact globally.

Supervisor's Signature: ______________________