DAYANANDA SAGAR UNIVERSITY

School of Engineering
Kudlu Gate, Bengaluru-560114

A Project Report

on

"DESIGN AND DEVELOPMENT OF GUIDED ROBOT FOR UNIVERSITY LOGISTICS"

Submitted in partial fulfilment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

HONNESH D [ENG20EC0036]

MANJESH S [ENG20EC0051]

NANDINI MURNAL [ENG20EC0056]

NAVANEETH S R [ENG20EC0058]

Under the guidance

of

Mr. PUNEETH S
ASSISTANT PROFESSOR,
DEPARTMENT OF ECE,
SCHOOL OF ENGINEERING, DSU

Department of Electronics and Communication Engineering,

Dayananda Sagar University, Bengaluru 560114

Academic Batch: 2020-2024 **Year: 2023-2024**

DAYANANDA SAGAR UNIVERSITY

School of Engineering

Kudlu Gate, Bengaluru-560114

Department of Electronics and Communication Engineering

CERTIFICATE

*This is to certify that the **Project** entitled **"DESIGN AND DEVELOPMENT OF GUIDED ROBOT FOR UNIVERSITY LOGISTICS"** has been successfully carried out by **HONNESH D [ENG20EC0036], MANJESH S[ENG20EC0051], NANDINI MURNAL [ENG20EC0056], NAVANEETH S R [ENG20EC0058] in** partial fulfilment of the requirement for the award of the degree **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING** by **DAYANANDA SAGAR UNIVERSITY** during the academic year 2022-23.*

Signature of Guide

MR. PUNEETH S
Assistant Professor
Dept. of ECE
SoE-DSU, Bengaluru

Signature of Coordinator

Dr.SARA K
Professor
Dept. Of ECE
SoE-DSU, Bengaluru

Signature of the Chairman

Dr. Arun Balodi
Dept. of ECE
SoE-DSU, Bengaluru

Signature of the Dean

Dr. Udaya Kumar Reddy K R
SoE-DSU, Bengaluru

External Examiner:

1) Name — Signature

2) Name — Signature

DECLARATION

*We hereby declare that the thesis entitled "*DESIGN AND DEVELOPMENT OF GUIDED ROBOT FOR UNIVERSITY LOGISTICS*" submitted to the Department of Electronics and Communication Engineering, School of Engineering, Dayananda Sagar University, Bengaluru, for the award of degree "Bachelor of Technology" in Electronics and Communication Engineering is a record of original work carried out by us under the guidance of MR.* PUNEETH S, Assistant Professor, *Dept. of ECE, Dayananda Sagar University, Bengaluru.*

To the best of our knowledge, this work has not been submitted for the award of any degree in any University or Institute.

Date :-13-05-2024

HONNESH D-ENG20EC0036
MANJESH S-ENG20EC0051
NANDINI MURNAL - ENG20EC0056
NAVANEETH S R -ENG20EC0058

ACKNOWLEDGEMENT

The completion of our Phase-1 final year project brings a sense of great satisfaction, but it is never complete without thanking the people responsible for its successful completion.

First, we take this opportunity to express our sincere gratitude to the Management School of Engineering, Dayananda Sagar University, for providing us with a great opportunity to pursue our bachelor's degree in this Institution.

In particular we would like to thank ***Dr. Udaya Kumar Reddy K R, Dean****, School of Engineering, Dayananda Sagar University for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Arun Balodi, Chairman,*** *Department of Electronics and Communication Engineering, Dayananda Sagar University, for providing the right academic guidance that made our task possible.*

We would like to thank our guide ***MR. PUNEETH S, Assistant Professor****, Department of Electronics and Communication Engineering, Dayananda Sagar University, for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our Project Coordinators ***Dr. Saara K & Dr. Mukti Chaturvedi,*** *Department of Electronics and Communication Engineering, Dayananda Sagar University and* ***all the staff members*** *of Electronics and Communication for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.

Signature of Students

ABSTRACT

Having a smart and helpful companion that can deliver your goods right to your doorstep. That's exactly what our delivery robot does. This innovative robot is designed to navigate to a pinpoint location, and it can be controlled from anywhere in the world. all you need to do is use your phone to communicate.

Once the goods are securely placed in the delivery robot, it is locked using the RFID mechanism. The delivery robot uses a camera module and object detection technology which is attached to the robot to visualize, and the goods is delivered safely to the customer at the location.

once the goods are loaded, the robot is ready to head to designated location. It uses GPS modules to pinpoint its exact location, making it easy to integrate with Google Maps for precise location and it can be tracked by the customer also.

But that's not all! The delivery robot is also equipped with a speaker, which facilitates communication between the customer and the robot. This feature ensures that any queries or issues can be resolved quickly and efficiently.

Our delivery robot is not only fast but also safe and friendly. It's like having a little bit of magic right at your fingertips, ensuring that you get your goods are quickly delivered in good condition. This technology represents the future of delivery, and it's here to make your life simple and more convenient.

CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

The delivery robots acts like super-smart delivery helpers that use fancy technology to bring your goods right to your door. These little machines are not your everyday gadgets, they're super clever! They use cool stuff like sensors (which help them see and avoid obstacles), cameras (to look around), and a smart brain called the Raspberry Pi 4 module to figure out the best way to travel through streets and buildings and get your goods to you without any hiccups.

New technologies, such as parachute-deployed probes and visual displays, are improving data collection. Despite the challenges, these drones play an important role in weather modeling and climate science by improving the accuracy of weather forecasts. Technological developments in recent years have paved the way for new solutions to many challenges.

What's neat is that these robots can keep your goods neat and safe as they zip around. That's one of the reasons why they're getting more popular – they're fast, and they don't get tired like people do. So, you can trust them to bring your goods to you in great shape!

These robots have special sensors that act like their eyes, helping them avoid things in their way. The Raspberry Pi 4 module is like their control center, making sure they move just right.

Adding cameras makes these robots even smarter. They use them to "see" and make quick decisions, like dodging people or cars during their journey. With the teamwork of sensors and cameras, these robots can travel on their own through streets and buildings, finding the quickest route to deliver your goods.

At the heart of all this tech is the Raspberry Pi 4 module – the robot's brain. This clever software processes information from sensors and cameras, making quick decisions. It helps the robot adapt to different places, plan the best routes, and make fast changes to make sure your goods arrive smoothly. The Raspberry Pi 4 module is like the boss, directing the robot's moves and decisions super efficiently.

A cool thing about these robots is how they keep your goods in great condition. They're designed to make sure your package stay safe during their journey. Unlike human delivery folks, these robots work tirelessly without getting tired, making them a reliable solution for meeting the growing needs of today's fast-paced consumers.

In conclusion, delivery robots are like high-tech wonders in the world of goods delivery. With their smart sensors, cameras, and intelligent software, they can navigate busy places with ease, ensuring it arrive on time. The Raspberry Pi 4 module is like their super-smart brain, showcasing the sophistication of these machines. As they become more popular, goods delivery robots are changing the way we get our packages, providing a reliable and innovative solution to keep up with the fast-paced demands of modern life.

1.1 Design

1.Chassis Construction:

• **Materials: Materials**: When developing the robot chassis, utilize robust, lightweight materials like carbon fiber or aluminum. These materials allow for simple movement within university buildings while maintaining durability.

• **Size:** Create a scalable chassis for the robot to accommodate various payload requirements and working surroundings. To be able to promote effective navigation across university buildings and pathways, the size should be optimized.

• **Inflation System:** Create an appropriate mobility system with wheels or tracks that can travel in multiple directions to enable easy movement across confined places and crowded regions.

2. Payload and Navigation:

Payload Capacity: Give the robot a customizable payload system that can carry various kinds of loads, like documents, lab equipment, and parcels, that are frequently seen in university supplies.

3. Energy:

• **Power Source:** The robot can run on a combination of hybrid power system or rechargeable lithium-ion batteries to provide it with sufficient power to last for extended periods of time between recharging periods.

• **Energy Management:** To minimize power consumptions and increase the robot's operating time between charges, utilize energy-efficient components and methods.

4. Communication and Control:

• **Wireless Connectivity:** To enable smooth data transfer between the robot and centralized control systems, install an appropriate wireless communication network (for example, Wi-Fi or Bluetooth).

• **Autonomous Navigation:** For the robot to operate independently, intelligent navigation methods and control software must be created. This involves route planning, recognition of obstacles, and flexible path adjustment.

• **Remote Monitoring:** Deliver a user-friendly interface that can be used on desktop computers or mobile devices to allow remote monitoring and control of the robot's actions.

5. Navigation system:

• **GPS module:** includes a GPS module for tracking and precise navigation

6.Customization and Flexibility:

• **Modular Design:** Use a modular design strategy to make it simple to integrate and customize extra features or sensors to fit with the distinct logistics needs of campuses.

7. Reasonable Costs:

• **Mass Production:** Robot are designed to facilitate mass production to achieve economies of scale and reduce construction costs.

• **Ready-to-use components:** Use ready-to-use and cost-effective components to reduce production costs.

8. Safety features:

• **Emergency lowering:** includes emergency procedure in case of unforeseen circumstances.

• **Fall detection:** The combined use of sensors and algorithms is required for fall detection and recovery, especially for situations involving gradients or uneven ground. Take precautions to prevent falls, such as modifying the robot's balance of gravity or turning on supporting systems.

9. Human-Robot Interaction:

•**Training and Support:** To help users and maintenance staff become confident and competent in using the robot, provide them with thorough training and support materials that cover operation, troubleshooting techniques, and safety precautions.

• **Collaborative Operation: Allow** the robot and human operators to collaborate on jobs that require human involvement or monitoring.

-Implement safety procedures and user permissions to promote safe and effective collaboration in shared places.

1.2. Working Principle

The guided robot for university logistics is powered by a combination of sensors, navigation algorithms, task management software, and communication networks. It begins by receiving instructions or tasks from a central control system, which could be a computer or an assigned operator. The robot detects its surroundings and identifies its location on campus using sensors such as cameras and distance sensors. It uses this information, together with pre-programmed maps or routes, to plot its course to complete the specified duties.

As the robot moves, it constantly collects data about its surroundings, identifying impediments and modifying its course to prevent collisions. This real-time navigation system assures safe and efficient transit throughout the campus. Furthermore, the robot connects wirelessly with the central control system, providing status updates, accepting new assignments, and reporting any problems found while operating. This communication enables flawless coordination between the robot and the control system, ensuring that tasks are executed precisely and efficiently.

To summarize, the working idea is that the guided robot navigates the campus autonomously, executing given tasks, avoiding impediments, and interacting with the central control system to ensure smooth and successful operation.
Here's a step-by-step explanation of how robots operate:

1. **Sensors:** RFID (Radio-Frequency Identification) is essential in allowing robots to interact with tagged items and assets in their surroundings. RFID technology uses electromagnetic fields to automatically detect, and track tags attached to things. In a university setting, RFID tags can be attached to equipment, books, documents, or even specific areas within buildings. When a robot outfitted with RFID sensors comes across an object or region with an RFID tag, it can wirelessly connect with the tag to obtain information such as the object's identity, position, or status. This functionality allows the robot to do a variety of activities, including inventory management, asset tracking, and delivering products to specific areas on campus.

• **Computer Vision Sensors:** Computer vision sensors, such as cameras and depth sensors, allow robots to sense and interpret visual information in their environment. These sensors take photos or video feeds of their surroundings, which are subsequently processed with computer vision algorithms to extract significant characteristics and objects. In the university setting, computer vision sensors enable robots to recognize and identify items, navigate complex surroundings, and interact with humans. For example, a robot outfitted with computer vision sensors may identify and avoid obstructions in its route, as well as recognizing students' or faculty members' faces or movements.

• **Obstacle Detection and Avoidance:** One of the key tasks of sensors in robots is to identify and avoid obstacles. Robots can detect and maneuver around obstacles in their path by integrating multiple sensors such as RFID and computer vision. RFID sensors detect static items or preset zones tagged with RFID tags, whereas computer vision sensors allow the robot to discern dynamic barriers such as people, furniture, or automobiles. The robot uses data fusion techniques to aggregate information from many sensors, creating a full map of its surroundings and planning collision-free routes to its objective. This skill is critical for assuring the safety and efficiency of robot navigation in busy or crowded areas, such as university campuses, where obstructions and pedestrians are prevalent.

• **Object Detection and Interaction**: In addition to obstacle avoidance, sensors allow robots to recognize and interact with items in their surroundings. RFID sensors help identify and track tagged things, allowing the robot to pick up, handle, and deliver items as needed. Computer vision sensors enhance this capability by providing visual feedback on the size, shape, and placement of things, allowing the robot to grip or move them precisely. This functionality is especially beneficial for jobs such as sorting library books, delivering parcels, or assisting with laboratory studies in the university setting.

2. USB camera and speaker:

• **Camera and Speaker Integration for Interaction:** Alternatively, the robot can use USB cameras and speakers to interact with its surroundings. Cameras on the robot capture live video streams of its surroundings, which are then routed to a central server for analysis. This live camera feed provides remote monitoring and surveillance, allowing university personnel to see the robot's operations in real time.

• **Two-Way Communication:** The robot's USB speakers enable two-way communication with anyone around it. Speech recognition technology allows the robot to reply to spoken directions or requests from faculty, students, and administrative staff. It can then respond vocally, providing information, directions, and help as needed. This feature enables smooth communication between the robot and its human counterparts, boosting collaboration and efficiency in the university setting.

• **Enhanced Navigation and Assistance:** By merging RFID technology with cameras and microphones, the robot can provide better navigation and help services. For example, if a lecturer needs help locating a specific classroom or office on campus, they can simply transmit their needs to the robot. The robot may then use its RFID capabilities to determine the lecturer's destination and direct them there using its integrated mapping and navigation system. Throughout the journey, the robot may deliver real-time updates and directions through its speakers, ensuring a smooth and efficient navigation experience.

• **Multi-Purpose Applications:** Beyond access control and navigation, the incorporation of RFID, USB cameras, and speakers offers up a plethora of potential applications for the university robot. For example, the robot can be designed to give campus tours to prospective students and visitors, offer educational presentations or announcements in lecture halls, or even facilitate remote guest lectures by allowing off-site speakers to interact with students via video conferencing. The robot's adaptability makes it a significant asset to the university community.

3. GPS Tracking:

GPS operates by utilizing a network of satellites orbiting the Earth. These satellites continuously emit signals that can be received by GPS receivers on Earth. By measuring the time, it takes for signals from numerous satellites to reach the receiver, as well as the precise location of those satellites, the GPS receiver can establish its own three-dimensional position (latitude, longitude, and altitude), as well as the current time.

GPS positioning accuracy varies depending on the receiver's quality, the number and position of satellites in view, and any obstructions such as buildings or trees. Under open sky conditions, current GPS receivers can often provide accuracy of only a few meters.

GPS has become an indispensable tool in a variety of fields, including aviation, maritime navigation, transportation, agriculture, outdoor enjoyment, and emergency services. Furthermore, GPS technology is frequently integrated into smartphones, automobiles, wearable devices, and other consumer electronics, allowing regular people to access location-based services and apps.

4. Power Source:

• **Rechargeable Battery System**: Robots deployed on university campuses use rechargeable battery systems to power their electrical components, which include CPUs (such as the Raspberry Pi 4), motors, sensors, and communication modules. Rechargeable batteries provide a dependable and portable power source, allowing robots to function autonomously for extended periods of time without the need for an external power supply. These batteries are often lithium-ion or lithium-polymer-based, chosen for their high energy density, lightweight design, and ability to power the robot's many functions.

• **Uninterrupted Functionality:** The usage of rechargeable batteries allows the robots' continuous usefulness, especially during important activities like package delivery or navigation aid. The batteries allow the robot to operate continually without downtime, maximizing its efficiency and productivity in the university setting. This continuous capability is critical for addressing the demands of time-sensitive tasks including delivering documents to staff members, directing visitors around campus, and supporting students with inquiries.

5. Security: Anti-Theft Alarms:

Robots can be outfitted with anti-theft alarms to prevent unauthorized removal or manipulation. Unauthorized entry attempts, physical impact, or tampering with the robot's components can all cause these alarms to go off. When activated, the anti-theft alarm generates loud alerts or visual signals to draw attention and notify nearby people of the security violation. Furthermore, the alarm can be coupled with the robot's communication system to transmit notifications to specific workers or security teams, allowing for quick response and intervention to avoid theft or vandalism.

• **RFID Blocking Technology:** To improve security, robots may use RFID blocking technology to restrict unwanted access to sensitive data or resources. RFID blocking methods prevent communication between RFID tags and readers, making it more difficult for hostile actors to clone or spoof RFID credentials and obtain illegal access to the robot or its connected systems. Robots can protect precious assets, secret data, or restricted areas on university campuses by using RFID blocking technology to ensure compliance with security rules and regulations.

• **Authentication and Access Control:** Authentication procedures and access control measures can be used to limit access to the robot and its functions. Users or operators may be asked to authenticate themselves using biometric credentials like fingerprints or face recognition, or by entering a secure PIN code or access card. Once authenticated, users are assigned certain privileges or permissions based on their role or degree of authorization. Access control regulations can be implemented at both the hardware and software levels, ensuring that only authorized users can interact with the robot, issue commands, or access sensitive data.

• **Remote Monitoring:** Robots with remote monitoring and surveillance skills can help with security oversight. Real-time monitoring of the robot's activities and surroundings is possible because of integrating cameras, sensors, and communication systems, which may be accessed via a centralized control center or mobile device. Security staff can remotely view the robot's live video feeds, sensor data, and status updates, allowing them to detect possible security risks, monitor suspicious activity, and respond quickly to security events as they occur.

• **Remote monitoring and surveillance**: It adds an extra layer of security to robots stationed throughout the university campus, complementing physical security measures and improving overall situational awareness. The operating premise of a smart bot is the seamless integration of sensors, data processing algorithms, and wireless connectivity to improve the shopping experience, inventory management, and bot delivery.

6. Customization and adaptability:

- **Modular design**: The guided robot's architecture is flexible, allowing institutions to tailor its functionality to their own needs. This versatility guarantees that the robot can adapt to a variety of tasks and situations on campus, whether by adding more sensors for environmental monitoring or implementing specific attachments for handling various types of goods.

- **Scalable deployment options**: The robot's design and versatility enable scalable deployment from small college campuses to large university complexes. Universities can tailor the number and configuration of robots to the size and complexity of their campus logistics operations, resulting in an efficient and cost-effective deployment regardless of scale.

- **Remote installation and control**: University logistics staff may easily install and control the guided robot using a simple software interface or mobile app. This remote-control feature enables operators to adjust navigation routes, task priorities, and communication settings from anywhere on campus, minimizing the need for human interaction and streamlining logistics management.

- **Collaboration with IoT devices**: The guided robot works smoothly with current Internet of Things (IoT) devices and campus infrastructure. This collaboration allows for data sharing and coordination between the robot and other campus systems, such as building management systems or inventory monitoring platforms, resulting in improved logistics efficiency and campus operations.

- **Innovation and continuous improvement**: The guided robot programmers are dedicated to ongoing innovation and development, ensuring that universities remain at the cutting edge of campus logistics technology. Colleges may successfully leverage the latest discoveries in robotics and automation by constantly refining software algorithms, hardware capabilities, and user interfaces to improve campus logistics performance and address growing operational concerns.

1.2. 1 IOT used in Robot

Developing a university-guided robot with IoT-enabled sensors gives up new opportunities for improving campus navigation and student services. Here's a breakdown of the primary components and considerations for such a system:

Sensor Selection: Select sensors that can navigate inside areas and detect obstructions, such as ultrasonic or infrared sensors. In addition, integrate sensors for collecting environmental data such as temperature, humidity, and air quality to ensure comfort and safety in university facilities.

Wireless Connectivity: Provide the robot with wireless connectivity, such as Wi-Fi or Bluetooth, so it can communicate with the university's network and other IoT devices. This allows for real-time data transmission and remote control of the robot's functions.

Data Transmission Protocols: Use efficient data transmission protocols to ensure that the robot communicates seamlessly with central servers or user devices. Protocols such as MQTT and HTTP can help share sensor data and control commands.

Power Supply: Give the robot a dependable power source, such as rechargeable batteries or a docking station, so it can operate continuously throughout the day. Energy-efficient designs and clever power management technologies can help the robot's battery last longer.

Navigation and Mapping: Use mapping and localization algorithms to help the robot navigate campus buildings autonomously. Use techniques such as SLAM (Simultaneous Localization and Mapping) to construct and update maps of indoor spaces in real time.

User Interface: Create a simple user interface for interfacing with the robot, such as a smartphone app or web portal. This interface should enable users to obtain assistance, provide navigation instructions, and access campus information services.

Integration with University Systems: the robot with current university systems and databases to gain access to important information such as class schedules, campus events, and facility locations. This connection allows the robot to deliver more individualized assistance to students, faculty, and visitors.

Safety Features: Implement safety elements to avoid crashes and ensure the safety of campus occupants. This could incorporate obstacle detection sensors, emergency stop buttons, and adherence to safety rules for robotic systems working in public areas.

CHAPTER 2

Literature Survey

In [1] “Real-time Object Recognition and Counting System for Smart Industrial Camera Sensors”. Shih-Hsiung Lee & Chu-Sing Yang. The incorporation of smart cameras equipped with GPUs can transform industrial inspection procedures in university guided robot projects that focus on automation and intelligence. Real-time object recognition and counting in high-speed industrial environments with massive data volumes is possible by utilizing GPU cores for parallel processing. This approach ensures efficient and precise inspection, paving the path for increased production capacity on manufacturing lines. Furthermore, the use of the NVidia Tegra TX1 platform with GPU CUDA cores and an ARM Cortex A57 processor, in conjunction with Basler USB 3.0 industrial cameras, enables a practical simulation of smart industrial cameras capable of complicated computations. The experimental data presented in this research validate the usefulness of The proposed paradigm provides a significant reference for future applications in the Internet of Things or Internet of Everything.

In [2] “Design of an Automated On-Demand Meal Delivery System Based on Emerging and Evolving Passenger Requirements” .The authors are Christopher P. Frank, Mathilde N. Deveraux, Rosemonde H. Ausseil, and Dimitri N. Mavris from the Georgia Institute of Technology in Atlanta, Georgia (30332-0150, United States). In a university guided robot project, some systematic procedures can be used to improve the robot's functionality and performance. This includes adopting novel solutions to identified issues in robotics applications. For example, the project could prioritize optimizing the robot's navigation system using advanced sensor technologies and algorithms, providing on-demand task execution capabilities via intuitive user interfaces or voice commands, and improving mobility and maneuverability in constrained environments by redesigning mechanical structures or incorporating adaptive locomotion mechanisms. Integrating feedback methods, like performance metrics or user assessments, Ensures ongoing improvement and alignment with project objectives. Careful consideration of aspects like power consumption, processing resources, and environmental limits, as well as extensive testing and validation, ensure that these advances are reliable and successful in real-world robotic applications.

In [3], An autonomous delivery robot to prevent the spread of coronavirus in the product delivery system. Murad Mehrab Abrar, Raian Islam, Md. Abid Hasan Shanto. The paper describes the creation of a delivery robot prototype intended to solve the issues provided by the coronavirus epidemic. The robot's goal is to provide safe and contactless delivery services by autonomously going to a certain location using GPS data. It includes a password-protected container to keep the delivery package secure, guaranteeing that only the intended recipient has access to it when it arrives. The robot can transport payloads weighing up to 1KG and can return to its starting point autonomously once the delivery is completed. This unique technology not only reduces the risk of infectious disease transmission, but it also provides a cost-effective strategy. Overall, the delivery robot prototype is a feasible and efficient response to the evolving demands of contactless delivery services during the epidemic.

In [4], “Object detection and separation. Using Raspberry Pi”.Sumeet Sanjay Walam, Bilal Sikandar Thakur, Siddhi Prakash Teli,Ravindra Ramchandra Nevarekar ,Mr. Suhas M. Patil.The creation of a low-cost touchscreen operant chamber based on the Raspberry Pi single-board computer provides a novel alternative for rodent cognitive testing. This approach can be modified for use in university-guided robots by incorporating similar touchscreen technology into robot platforms intended for behavioral testing or cognitive assessment tasks. University robots can be outfitted with touch-sensitive screens and interactive stimuli to conduct activities that mimic human neuropsychological examinations like decision-making, attentiveness, and spatial learning. For example, university-guided robots can use touchscreen interfaces to show visual signals or interactive tasks to experimental subjects like rats or other animals. These tasks may These tasks may involve operant-based behavioral paradigms in which the participant must respond to specific signals or stimuli displayed on the touchscreen in order to gain rewards or incentives. The robot can independently regulate task presentation, record subject replies, and provide rewards, resulting in quick and uniform data collecting.

Furthermore, the Raspberry Pi-based system's inexpensive cost makes it suitable for university research labs with limited resources. Researchers can create and configure their own touchscreen-operated robot platforms to meet their individual experimental requirements, rather than relying on pricey commercial systems. This flexibility allows universities to broaden their skills in behavioral and cognitive research while maximizing resources.

In [5], Remotely accessible smart lock security system with key features. The abstract by Sambasiva Rao Pinjala1 and Shreya Gupta2 describes the development of a cost-effective touchscreen operant chamber for rodent cognitive testing using the Raspberry Pi single-board computer. This chamber enables tasks comparable to those enabled by cutting-edge systems such as the Cambridge Neuropsychological Test Automated Battery. It allows mice to interact with visual stimuli via a touch-sensitive screen, eliciting reactions through nose pokes in order to collect rewards. This chamber offers a low-cost alternative to expensive commercial technologies, allowing researchers with limited funds to conduct advanced cognitive testing on preclinical animal models.

2. 1 Problem Statement

Develop a delivery robot system capable of efficiently and safely transporting packages from a central distribution point to specified destinations within urban environments, addressing challenges such as navigation in crowded areas, obstacle detection and avoidance, real-time tracking, security, and energy efficiency

We intend to create and implement a sophisticated delivery robot system designed specifically for metropolitan situations. Our technology will be designed to transfer items efficiently and securely from a central distribution center to specific locations inside congested cities. To accomplish this, we will address a number of urban-specific challenges, such as navigating densely populated areas, detecting and avoiding obstacles in real time, ensuring precise package tracking, prioritizing security measures, and optimizing energy consumption for sustainability. Our delivery robot system will transform last-mile logistics by combining cutting-edge technology and robust algorithms, increasing ease and dependability in urban package delivery.

2.2 Objectives

• **Efficient Delivery:** Food delivery robots aim to streamline the delivery process, reducing delivery times and increasing the number of orders a service can fulfill in a given time frame. By
leveraging advanced technologies, these robots can navigate through various environments and traffic conditions to reach their destinations promptly.

• **Time Efficiency:** The robot should aim to complete deliveries within a specified time frame, minimizing delays and ensuring customer satisfaction.

• **Safety:** The priority is safety by implementing obstacle detection mechanisms to prevent collisions with pedestrians, vehicles, and other things.

• **Package Security:** Create a secure package storage to protect deliveries from theft, damage, or
environmental factors.

• **User-Friendly Interaction:** Designing a user-friendly interface for customers to interact with the robot, track their food deliveries, and receive notifications.

• **Accessibility**: Ensure that the delivery robot system is accessible to all members of the community, including individuals.

• **Compliance with Regulations**: The robot will adhere to all relevant laws and regulations related to robot operations, traffic, and safety.

• **Cost-effectiveness**: The deployment of food delivery robots seeks to optimize operational costs
associated with traditional human-based delivery services. Robots can operate continuously without breaks, reducing labor costs and improving the overall economic feasibility of food delivery businesses.

• **Reliability:** Food delivery robots are designed to provide a consistent and reliable delivery service. They eliminate issues related to human factors such as fatigue, variability in performance, and potential errors, ensuring that deliveries are made with a high level of accuracy and dependability.

• **Customer Satisfaction:** The ultimate goal is to enhance the customer experience. Food delivery
robots aim to provide a convenient and contactless delivery option, meeting the expectations of modern consumers who prioritize efficiency, speed, and safety in their delivery services.

• **Adaptability to Various Environments:** These robots are engineered to navigate diverse environments, including urban streets, building interiors, and public spaces. The ability to adapt to different situations allow them to serve a wide range of customers and businesses effectively

CHAPTER 3

Proposed methodology

3.1 proposed design workflow

1.Project scope and needs analysis:

1. Determine project scope, objectives, and logistic needs.

2. Check specific environmental conditions for maintenance.

3. Identify requirements for affordable technology.

2. Sensor selection and combination:

1. Choose appropriate sensors to collect data.

2. Combine the sensor with microcontroller and use the communication protocol.

3. Airship design and simulation:

1. 3D modeling and simulation using Catia.

2. Ensure efficiency, structural integrity and payload capacity.

4. Prototype development:

1. Create a plane according to your design.

2. Combine microcontrollers and sensors to create data processing software.

5. Testing and Calibration:

1. Conduct environmental testing and testing.

2. Check the robot's stability, maneuverability and overall performance.

6. Analysis and interpretation of data:

1. Improved real-time data analysis algorithms.

2. Use visualization tools and analyze patterns to make decisions.

7. Optimization and iteration:

1. Identify areas for improvement based on testing and analysis.

2. Optimize the robot's design, sensors or software algorithms to improve performance.

8. Documentation and Creative Commons:

1. Design documents, code and test scores.

2. Create user manuals and share research results with the scientific and agricultural communities.

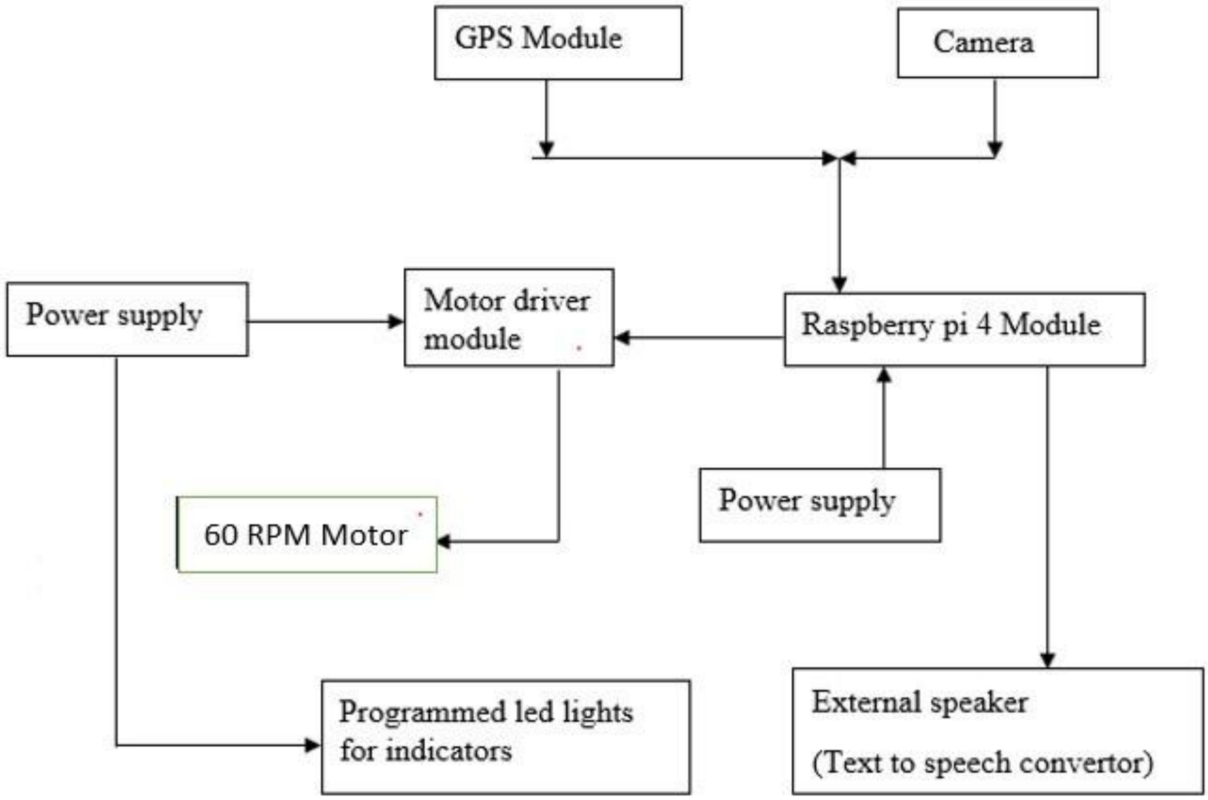


Fig3.1Design Workflow

3.2 Project Requirements

3.2.1 Software features:

1. Accurately document project-specific requirements.

2. The feasibility study is based on the requirements collected to test the feasibility of the project.

3. Modularize projects as needed.

4. Architectural and detailed design is drawn for each module.

5. During the study, comments are used to explain the result, which is important for knowledge and future use.

6. Extensive testing is performed to verify that the project is as designed.

3.2.2. Raspberry Pi 4 :

The Raspberry Pi 4 is the fourth edition of the popular single-board computer series. Developed by Raspberry Pi Trading, it introduced several enhancements over its predecessor, including an upgraded system-on-chip (SoC) to the Broadcom BCM2711, the incorporation of USB 3.0 in two of its four USB ports, and the availability of larger RAM configurations than the previous Raspberry Pi 3 series' standard 1 GB. Unlike previous models, the Raspberry Pi 5 was introduced as a successor to the Pi 4.

Here are some other features of the Raspberry Pi 4:

• Bluetooth 5.0

• Gigabit Ethernet

• Two micro-HDMI ports that support 4k@60p

• 2 lane MIPI DSI display port

• 2 lane MIPI CSI camera port

• 4-pole stereo audio and composite video port

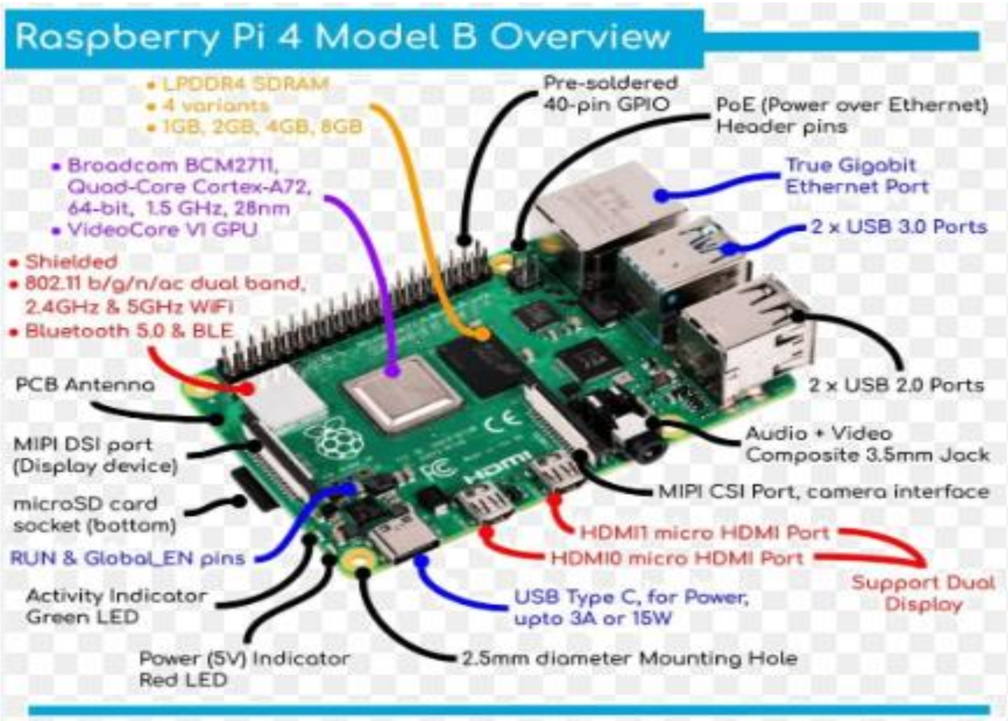


Fig3.2 Raspberry Pi 4 module

Key Features:

• **CPU:** The Raspberry Pi 4 was powered by a Quad-core ARM Cortex-A72 CPU clocked at 1.5 GHz or 1.8 GHz, which boosted processing capabilities over its predecessors. The newer C0-stepped chip replaced the original B0 stepping, resulting in improved thermal performance.

• **GPU:** The Broadcom BCM2711 SoC had an improved VideoCore VI GPU clocked at 500 MHz, delivering better graphics performance than previous models. This GPU also included a memory manager, which allowed for additional memory and compatibility with the OpenGL ES 3.1 and Vulkan 1.2 standards.

• **RAM:** The Raspberry Pi 4 offered consumers the option of 1 GB, 2 GB, 4 GB, or 8 GB of LPDDR4 RAM clocked at 3200 MHz, giving them more memory flexibility to meet their computing needs.

• **IO and Connectivity:** Significant upgrades were made to IO capabilities, including the update of USB ports to include two USB 3.0 ports and two USB 2.0 ports, as well as gigabit Ethernet compatibility enabled via a dedicated controller on the SoC. The previous single HDMI port was replaced with two micro-HDMI connectors, and Bluetooth was updated to version 5.0. In addition, the power supply connector was upgraded from Micro-USB to USB-C, and IEEE 802.3af Power over Ethernet (PoE) was implemented.

• **Compute Module 4:** The Compute Module 4 is geared at embedded and industrial applications, with RAM options comparable to the Pi 4 and the addition of eMMC onboard storage. The Compute Module 4 used 100-pin connectors for IO, exposing the PCIe 2.0 bus for greater connectivity and versatility.

• **Raspberry Pi 4:** The Raspberry Pi 4 incorporates a Pi 4 into a keyboard form factor, making it ideal for desktop PC use. It was available either standalone or as part of a desktop kit, and it included 4 GB of RAM as well as a variety of peripherals for easy setup.

• **Peripherals:** Notable peripherals included the Compute Module 4 IO Board, which improved interface access and addressed USB-C power difficulties that had been reported in the first Pi 4 release. The implementation was improved in version 1.2, allowing for interoperability with high-speed USB-C cables.

• **Video Output:** Dual-monitor systems using micro-HDMI ports can achieve resolutions of up to 4K. This makes it suitable for a variety of multimedia applications, including desktop computers, media centers, and digital signage.

• **USB connections:** The Pi 4's two USB 2.0 and two USB 3.0 connectors allow for faster data transfer rates than previous generations. These ports are suitable for connecting keyboards, mice, external storage devices, and other peripherals.

• **GPIO Pins:** The Pi 4 has quicker data transmission speeds than previous models due to its two USB 2.0 and two USB 3.0 ports. These ports let you connect keyboards, mouse, external storage devices, and other peripherals.

• **Support for Various Operating Systems:** The Pi 4 can run Ubuntu, Raspbian, Raspberry Pi OS, and other Linux-based operating systems. It also supports a range of operating systems and third-party programs.

• **Form Factor**: The Raspberry Pi 4 is ideal for robotics, education, embedded projects, and Internet of Things (IoT) applications since it retains the smaller form factor of prior iterations. Improved Speed: Equipped with a faster CPU and extra memory, the Raspberry Pi 4 performs better for demanding jobs.

• **USB connections**: The Pi 4 has two USB 2.0 and two USB 3.0 connectors, allowing for faster data transmission than previous models. Users can attach a variety of peripherals to these ports, such as external storage devices, keyboards, and mice.

• **Enhanced Performance:** The Pi 4 outperforms its predecessors in terms of computing capacity. It is powered by a 64-bit, quad-core Cortex-A72 (ARMv8-A) Broadcom BCM2711 CPU, which enhances multitasking and performance.

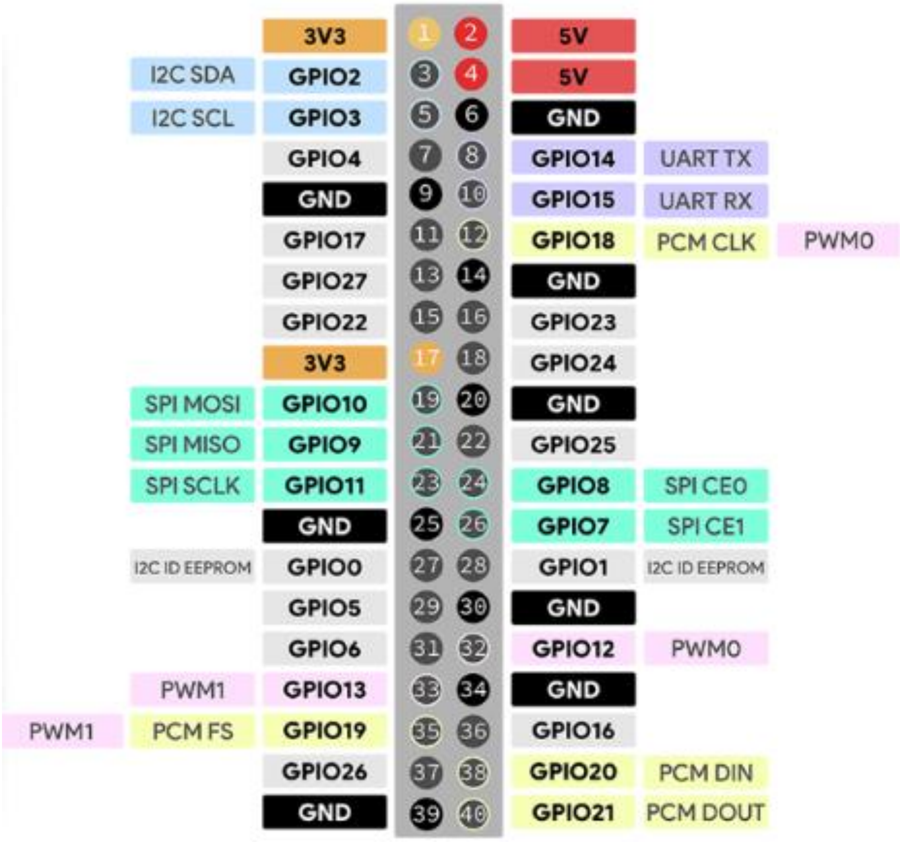


Fig3.3 GPIO Pins

3.2.3 Arduino Nano

Arduino is an open platform for electronic projects that includes software development environments (IDEs). It's simple to use, simply takes a USB cord to upload the code, and is written in C++. The prototype design streamlines the microcontroller's operation. The Arduino community develops microcontroller-based modules. In contrast to AVR Studio's assembly language, the Arduino IDE is developed in C, which allows access to a vast library.

• **Extensive community support:** The Arduino platform has a huge and active community of developers, amateurs, and professionals that exchange projects, tutorials, and problem-solving tips. This thriving network encourages collaboration, knowledge exchange, and innovation in energy development, including blimps.

• **Cross-platform compatibility:** Arduino Nano works with a wide range of operating systems, including Windows, macOS, and Linux, making it accessible to a large number of users. This cross-platform connectivity enables developers to use their preferred workflows.

• **Integrated Development Environment Nano:** Arduino Nano is a simple and user-friendly environment for writing, compiling, and exporting code for Arduino compatible microcontrollers. The simple UI and built-in code editor simplify development for beginners and unskilled users.

• **Hardware Compatibility:** Arduino Nano is compatible with a wide range of Arduino microcontroller boards, including popular models like the Arduino Uno, Ide, and Mega. These boards are large and affordable, making them ideal for use in airships.

•**Software Compatibility:** Arduino Nano employs a simple C++ programming language model that is appropriate for novices and those with limited programming knowledge. This syntax is simple to learn and understand, leading to increased speed and efficiency.

• **Plug-and-play functionality:** The Arduino board is simply connected to a computer by USB cable, allowing for seamless program code loading and debugging. The Arduino Nano's plug-and-play feature simplifies installation and removes the need for sophisticated hardware configuration, making it perfect for rapid prototyping in airships.

• **Extensive Libraries**: Arduino Nano supports a large number of pre-written codes and libraries, which contain a variety of functions and devices. These libraries simplify activities like connecting with sensors, manipulating actuators, and communicating with external devices, saving time and effort during development.

• **Customization and Expansion**: Arduino Nano is an open-source platform that allows users to customize and expand its capabilities to meet their individual needs. This openness fosters creativity and collaboration throughout the Arduino community, resulting in the creation and refinement of new airship features.

• **Integration with Sensors and Modules**: Arduino Nano allows integration with a wide range of sensors, modules, and peripherals used in aeronautical applications. Developers may easily attach temperature, humidity, GPS, and wireless communication modules to capture and send weather data from ships to ground stations or computers.

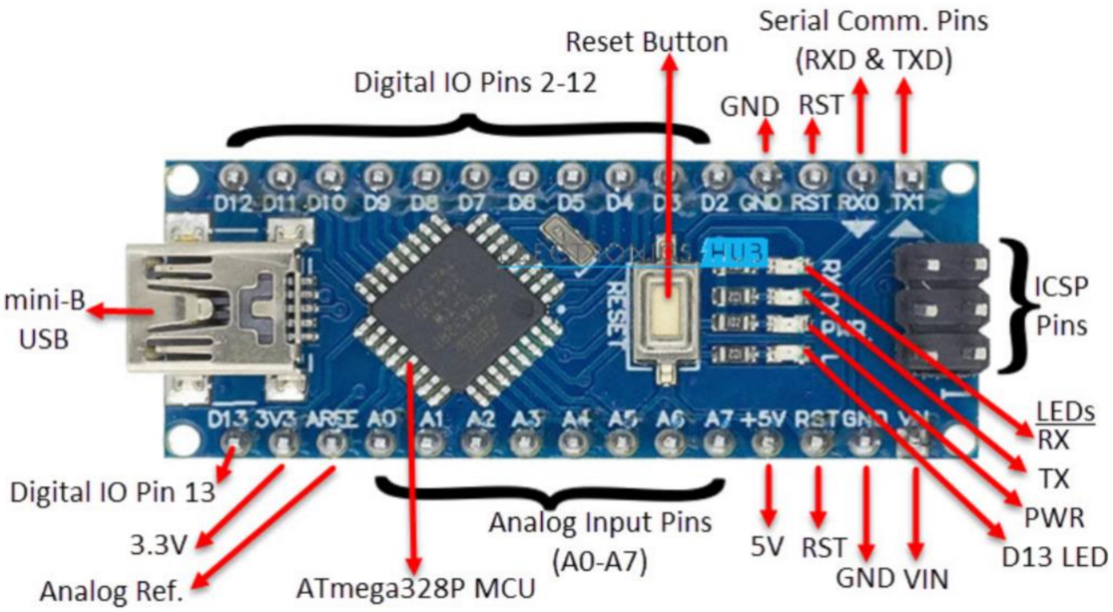


Fig3.4 Arduino Nano

3.2.4 Real VNC

RealVNC (Virtual Network Computing) is a software program and remote access technology that enables users to access and operate computers or devices via a network. Users may view the desktop interface and interact with the remote computer as if they were physically there in the same area.

- **Software Application:** RealVNC offers a variety of remote access and control software applications, such as VNC Server, VNC Viewer, and VNC Connect. These apps are available for a variety of operating systems, including Windows, macOS, and Linux, as well as mobile platforms like iOS and Android.

- **VNC Server**: The VNC Server software is installed on the computer or device that users wish to access remotely. It operates in the background and connects distant users to the host computer over a network. The VNC Server captures the desktop interface and transmits screen updates to the linked VNC Viewer clients.

- **VNC Viewer**: The VNC Viewer software is installed on the client devices from which users want to connect to and operate the remote computer. It has a graphical user interface (GUI) that allows users to connect to the VNC Server, view the remote desktop, and interact with the host computer via mouse and keyboard input.

- **VNC Connect**: RealVNC offers a subscription-based service called VNC Connect that includes extra features and functionality for remote access and administration. It provides cloud connectivity, encryption, authentication, and deployment solutions to enterprises and organizations.

- **How It Works**: - Connection Establishment To create a remote connection, the VNC Viewer client sends a connection request to the VNC Server on the distant computer. The request provides the remote computer's IP address or hostname, as well as a secure authentication token.

- **Desktop Sharing**: Once the connection is established, the VNC Server captures the remote computer's desktop interface and encodes screen changes as compressed data packets. It delivers these packets across the network to the linked VNC Viewer client.

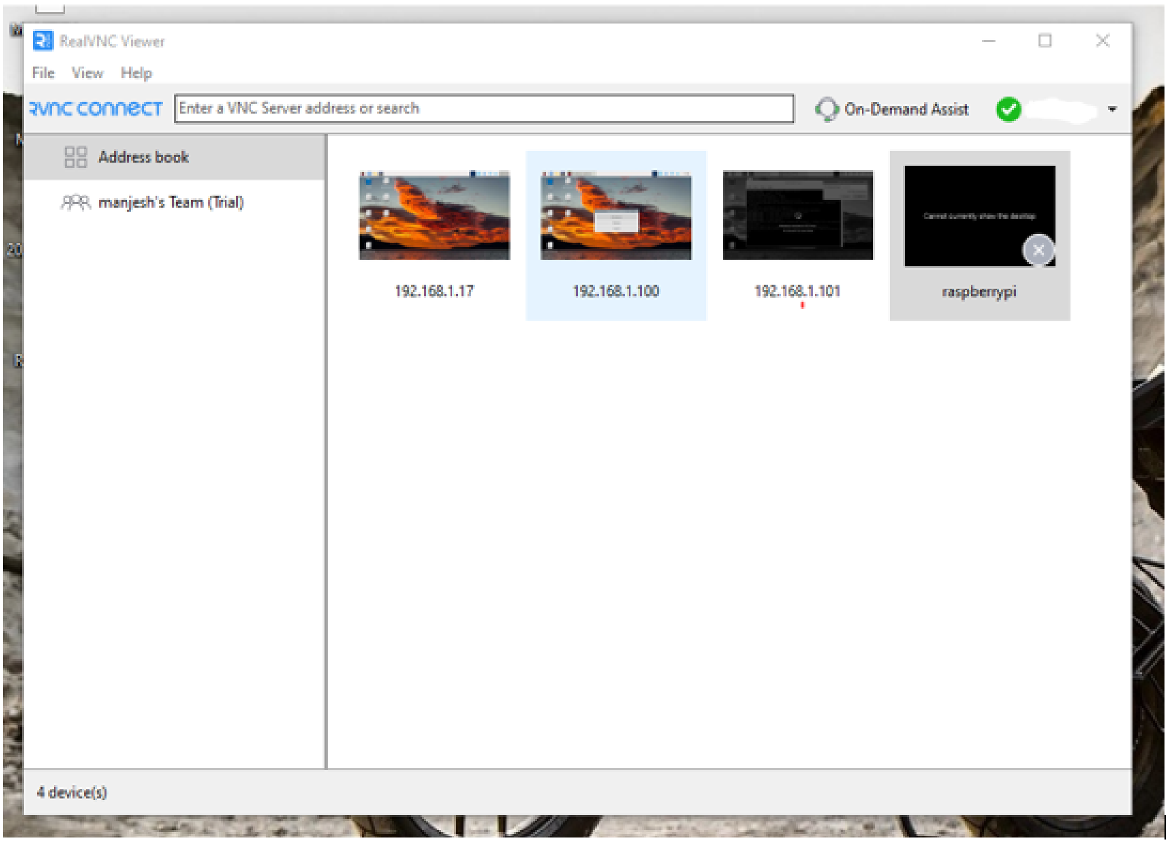
- **Remote Control**: The VNC Viewer client accepts data packets from the VNC Server and decodes them to recreate the remote desktop interface. The VNC Viewer interface allows users to interact with the remote computer by moving the mouse pointer, clicking on icons, inputting text, and doing other operations.

- **Bi-Directional Communication**: RealVNC supports bi-directional communication between the VNC Server and VNC Viewer, enabling users to transfer files, talk, and cooperate in real time during remote sessions.

- **Security and Encryption**: RealVNC employs encryption and secure authentication procedures to safeguard remote connections against unwanted access and data breaches. It uses industry-standard encryption methods, such as AES (Advanced Encryption Standard), to protect data transfer across the network.

- **Cross-Platform Compatibility**: RealVNC provides cross-platform remote access, allowing users to connect from a variety of operating systems and devices, including Windows, macOS, Linux, iOS, and Android.

- Businesses, IT professionals, support teams, and individuals utilize RealVNC for a broad range of remote access and support applications, including IT administration, remote troubleshooting, software development, education, and telecommuting. Its ease of use, dependability, and security features make it a popular choice for remote desktop access and control across a wide range of industries and applications.



RealVNC software

3.2.5 RFID lock system

- **RFID Reader:** This gadget detects and reads RFID tags and communicates with the Arduino Nano using common communication protocols such as UART or SPI.

- **RFID Tags:** These little electronic devices include unique IDs and are recognized by the RFID reader when brought into close proximity.

- **Arduino Nano:** The Arduino Nano serves as the system's central processing unit, receiving data from the RFID reader, interpreting it, and orchestrating actions, most notably regulating the locking mechanism.

- **Locking Mechanism:** This crucial component locks or releases the lock in response to instructions from the Arduino Nano. Examples include electronic solenoid locks and motorized bolts.

- **Power Supply:** For the system to function properly, it requires a consistent power source, which can be obtained from a battery, or an external power supply connected to the Arduino Nano.

- **User Interface:** This module, which allows users to interact with the system, may include elements such as keypads or screens for entering PIN numbers or displaying system status.

- **Feedback Mechanism:** This optional component, which provides users with real-time feedback on the lock's state, may include LEDs to indicate locking status or an audio buzzer for alarms.

- **Communication Interface:** If integration with other devices or a centralized control system is required, the system may include communication interfaces such as UART or SPI, which allow the Arduino Nano to communicate with other devices.

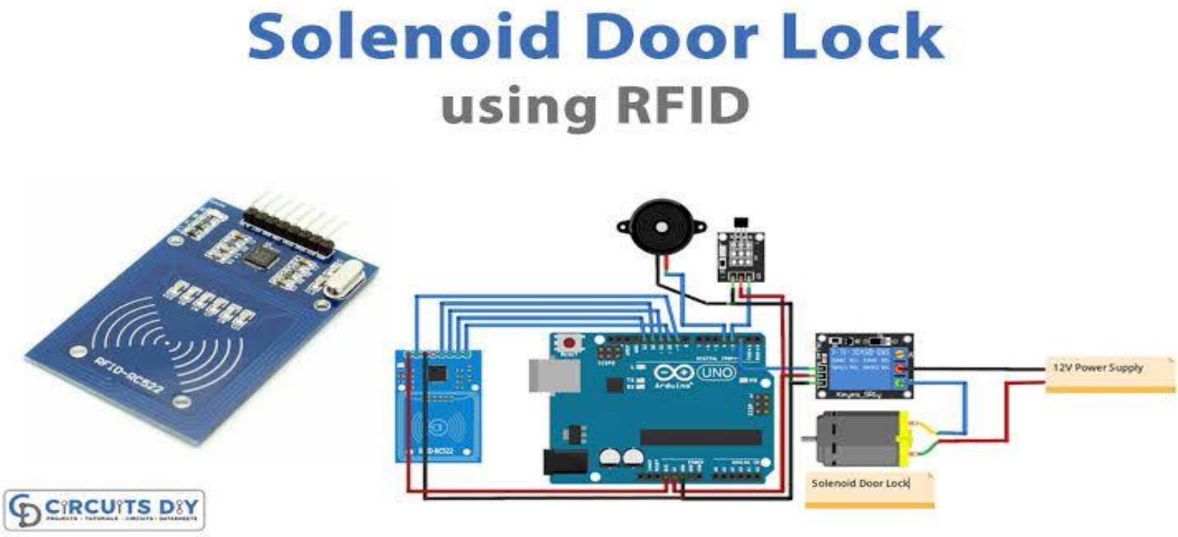


Fig3.5 RFID Lock System

3.2.6 Web requirements

Programing Languages: C and C++

The Raspberry Pi, like the ESP32, offers a versatile platform for embedded development, and programming it often involves languages like C and C++.

C serves as the primary programming language for the Raspberry Pi, offering direct access to its hardware and peripherals. Its efficiency and flexibility make it well-suited for tasks requiring optimized performance and low-level control. Additionally, C's syntax is relatively straightforward, making it accessible to those new to programming.

C++ is another viable option for Raspberry Pi development, building upon C with additional features such as object-oriented programming. This makes it suitable for larger and more complex projects, providing a modular and organized approach to code structure.

The Raspberry Pi supports both C and C++ compilation and execution, allowing developers to fully utilize its hardware for a variety of applications such as IoT projects, robotics, and multimedia.

Finally, the choice between C and C++ for Raspberry Pi programming is determined by project needs, language familiarity, and the desired level of abstraction and organization in the codebase.

3.2.7 3D printing

The base model was 3D printed on the Creality Ender 3 V3 SE 3D printer. The printer uses fused deposition modeling (FDM) technology, in which molten plastic filaments are extruded layer by layer to form the desired 3D object. This is a low-end FDM 3D printer that can print with PLA filament and other materials. The technical specs include: -

Printing material: 1.75mm (about 0.068 in) thick PLA filament.
Printing temperature range: 190-205 °C.
Bed temperature: 60 °C

Creality print 5.0 Software was utilized for the original design phase of the Robot, as well as for simulation. This enables virtual testing and chassis structural improvement, all using Catia software.

Catia is a powerful 3D computer animation, modeling, simulation, and visualization software created by Autodesk. It is commonly utilized in the film, television, and video game industries to produce high-quality 3D films, visual effects, and virtual environments. Simple model. The cuboid base, top ring, and connecting cylindrical rods will all be designed using software design tools. Catia's 3D modeling, texturing, and rendering capabilities facilitate virtual prototype and infrastructure visualization prior to 3D printing. Make an exact 3D model of the basic model with the dimensions

Product design and Construction: -

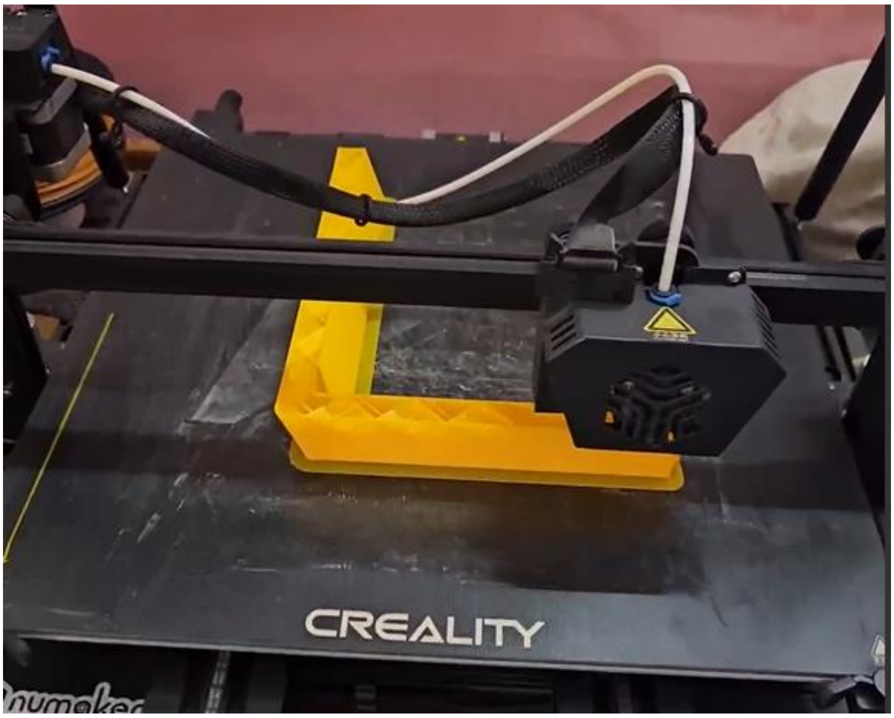


Fig 3.6 3D printing

3.2.8 Details of the components:

1. Raspberry Pi 4:

The Raspberry Pi is a small single-board computer. By adding accessories such as a keyboard, mouse, and display to the Raspberry Pi, it can function as a little personal computer.

Raspberry Pi is widely used for real-time image and video processing, IoT-based applications, and robotics applications.

Raspberry Pi is slower than a laptop or desktop computer, yet it nevertheless provides all the required functions or abilities while using very little power.

The Raspberry Pi Foundation officially supplies the Raspbian operating system, which is based on Debian. They also provide the NOOBS operating system for the Raspberry Pi. We can install a variety of third-party operating systems, including Ubuntu, Arch Linux, RISC OS, and Windows 10 IOT Core.

Raspbian OS is an official operating system that is accessible for free use. This operating system has been efficiently optimized for usage with the Raspberry Pi. Raspbian offers a graphical user interface that contains tools for browsing Python programming, office, and gaming.

To store the operating system, we need to use SD cards (a minimum of 8 GB is recommended). Raspberry Pi is more than just a computer because it allows you to design applications using the on-chip hardware, known as GPIOs. Accessing GPIO allows us to connect and control devices such as LEDs, motors, and sensors.

Raspberry Pi processors

It is equipped with an ARM-based Broadcom Processor SoC and an on-chip GPU.

The CPU speed of the Raspberry Pi ranges from 700 MHz to 1.2 GHz. It also includes on-board SDRAM with capacities ranging from 256 MB to 1 GB. Raspberry Pi also includes an on-chip SPI, I2C, I2S, and UART module.

2. 6ORPM DC stepper motor:

When we say, "60 rpm gear motor," we mean a specific sort of motor designed for precise and controlled rotational action. These motors are widely used in a variety of industries and applications that require continuous and reliable rotation.

Gear motors are essential components in applications such as robotics, automation, conveyor systems, and mechanical assembly that require precise movement. Their ability to provide continuous torque and rotating speed makes them excellent for activities that need precise motion.

Now, let's go deeper into what "60 rpm" means. The "rpm" stands for revolutions per minute, which is the rate at which the motor's output shaft completes one full rotation in a minute. In this scenario, a 60-rpm gear motor makes a full spin every 60 seconds, ensuring steady and predictable motion.

What distinguishes gear motors is the introduction of gears into the motor assembly. These gears reduce the output speed of the motor from its high spinning speed to the desired low speed. These motors can handle heavier loads with ease because they use gear ratios to increase torque while decreasing speed.

This torque amplification is especially useful in cases when precise control and adjustment of rotational motion are required. Whether it’s smoothly moving conveyor belts, precisely placing robotic arms, or managing the movement of mechanical components in assemblies, the 60-rpm gear motor provides the dependability and precision required to do these tasks successfully.

The "60 rpm gear motor" represents a specific engineering solution designed to suit the needs of applications that require regulated steady rotational motion. Its gear integration allows for precise speed reduction and force amplification, making it a critical component in a wide range of industrial and mechanical applications.

Fig 3.7 60 RPM DC stepper motor

3. Driver module:

The L298N motor driver module is a popular and adaptable dual H-bridge motor driver integrated circuit (IC) module that is widely used in robotics, automation, and DIY electronics projects. It provides a simple solution for controlling the speed and direction of DC motors, making it popular in a variety of applications that require motor control.

1. H-Bridge Configuration: The L298N module has two H-bridge circuits, which allow it to independently regulate the speed and direction of two DC motors. Each H-bridge consists of four transistors organized in a way that provides bidirectional control of motor rotation.

2. Voltage and Current Rating: The L298N module normally accepts a wide range of input voltages, ranging from 5V to 35V, making it suitable for a variety of power sources, including batteries and power supplies. It can withstand currents of up to 2A per channel (with sufficient heat sinking), allowing it to power a variety of DC motors.

3. Control Inputs: The module accepts control signals from a microcontroller or other control circuits to control the speed and direction of the attached motors. Each motor normally requires four control signals: two for direction (forward or backward) and two for speed (by pulse-width modulation, or PWM).

4. Built-in Diodes: The L298N module has built-in freewheeling diodes (also known as flyback diodes) that safeguard the circuit from voltage spikes caused by the motor when it is turned off. These diodes protect the module from harm while also ensuring smooth motor operation.

5. Heat Sink: The L298N module often includes a built-in heat sink or a mounting hole for an external heat sink. This helps to disperse heat created during motor running, enabling consistent and continuous functioning, particularly at higher currents.

6. Versatility: Because of its rugged construction and versatility, the L298N motor driver module is employed in a variety of applications, including robot platforms, RC cars, CNC machines, and numerous motorized projects.

The L298N motor driver module is a popular choice among amateurs and electronics enthusiasts because of its simplicity, dependability, and adaptability in controlling DC motors. Its twin H-bridge architecture, wide input voltage range, and current-handling capabilities make it ideal for a variety of motor control applications.

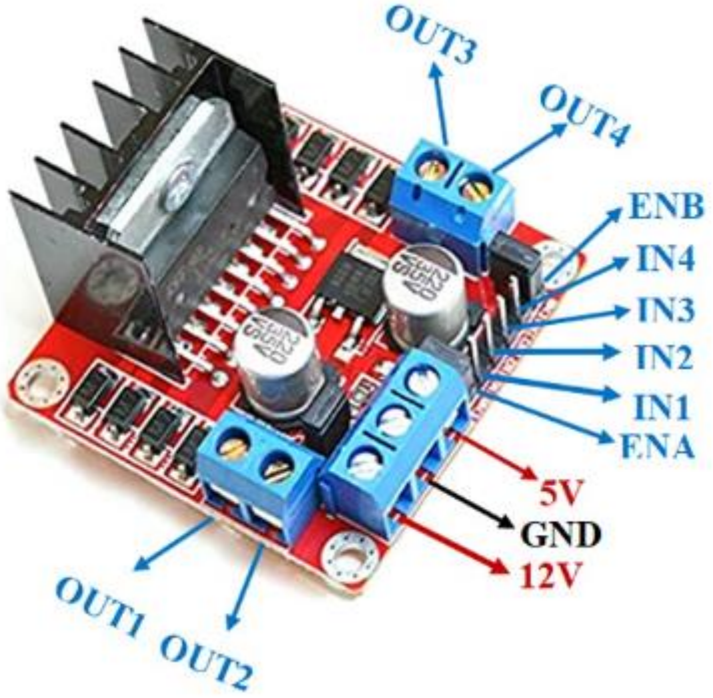


Fig3.8 Driver module

4. Wheels:

Wheels are vital components in a wide range of industries, including transportation and automotive, robotics, equipment, and everyday consumer products. They act as crucial interfaces between a moving object and its supporting surface, providing mobility, stability, and control. Here are some important features and data concerning wheels:

Wheel Components: A conventional wheel has multiple components, including:

- Tire: The outer component of a wheel that makes contact with the ground or surface. It improves traction, absorbs shock, and protects the wheel.

- Rim: A circular metal or composite structure that supports the tire and serves as the axle's mounting point.

- Spokes: In spoked wheels, spokes connect the rim to the hub, providing structural support and distributing loads evenly across the wheel.

Wheels can be made from a variety of materials, depending on the application needs.
Common materials include:

- Plastic: Plastic wheels are lightweight and inexpensive, making them ideal for situations where weight and cost are important considerations.

- Rubber: Rubber tires offer great traction, shock absorption, and cushioning, making them suitable for pneumatic and solid wheels in a variety of applications.

Design Considerations: When choosing wheels for a particular application, consider weight capacity, operating environment (inside or outdoor, difficult terrain, etc.), traction needs, speed, and durability.

Fig 3.9 Wheel

5. Speaker:

A speaker is a device that converts electrical impulses to sound. It has a few major parts:

Driver: This is the portion that generates sound. It's like a small motor that goes back and forth swiftly to push air and produce sound waves.

Enclosure: The driver sits in a box or enclosure. This box improves the sound quality by adjusting how the air travels around the driver. There are several enclosures for different types of sounds.

Crossover: A crossover is found within speakers with various drivers (such as one for low and one for high frequencies). It functions similarly to a traffic officer for sound signals, sending the correct sounds to the appropriate driver.

Terminals: These are the plug-in points for wires from your amplifier. They ensure the electrical signal reaches the driver.

Grille: This is simply a protective shield for the driver. It prevents items from poking or getting inside while still allowing sound to get through.

Port: Some speakers contain an opening or tube known as a port. It helps with some types of sounds, particularly low bass tones.

Fig 3.10 Speaker

6. Display:

The Raspberry Pi 4's display capabilities enable it to send video signals to a variety of screens, making it suitable for a wide range of applications like digital signage, multimedia, gaming, and desktop computing. Here are some key features and details concerning the Raspberry Pi 4's display capabilities:

1. Video Output Ports: The Raspberry Pi 4 contains two micro-HDMI ports, allowing for up to two display outputs. These ports support HDMI displays such as monitors, televisions, projectors, and other HDMI-compatible devices.

2. Resolution Support: The Raspberry Pi 4's micro-HDMI connectors can handle resolutions of up to 4K (3840x2160 pixels) at 60Hz, resulting in crisp, high-definition video output. This produces vivid, detailed images, making it excellent for multimedia and gaming applications.

3. Display Configuration: The Raspberry Pi 4's dual micro-HDMI connectors allow it to power two screens at once, each with a resolution of up to 4K. This capability is handy for multitasking,

4. Touchscreen Support: In addition to typical HDMI monitors, the Raspberry Pi 4 supports a variety of touchscreen displays and accessories, allowing for interactive applications and user interfaces. Touchscreen screens can be linked via HDMI or the Raspberry Pi's Display Serial Interface (DSI), providing more options for various applications.

5. Customization and Expansion: The Raspberry Pi community offers a diverse range of display accessories and expansion boards, including LCD displays, OLED displays, e-ink displays, and HATs (Hardware Attached on Top), allowing users to tailor their display capabilities to specific project requirements.

The Raspberry Pi 4's display abilities make it a versatile and powerful platform for a wide range of applications, including basic desktop computing, video playback, digital signage, and interactive

The Raspberry Pi 4's display capabilities make it a versatile and powerful platform for a wide range of applications, such as basic desktop computing, multimedia playback, digital signage, and interactive installations. It enables high-resolution screens, dual display output, and touchscreen interfaces, offering customers more control and personalization over their projects.

Fig 3.11 Raspberry Pi Display

7. Object Detection Sensor (Ultrasonic sensor)

Ultrasonic sensors are widely employed in a variety of applications, including object identification, distance measuring, and obstacle avoidance. Here is a summary of ultrasonic sensors and their components:

Transducer: The transducer turns electricity into sound waves (ultrasonic waves) and back again. It looks like a little speaker and microphone combined.

Transmitter: It emits ultrasonic waves into the air or space surrounding the sensor.

Receiver: The receiver listens for ultrasonic waves that bounce back after hitting an item.

Control circuitry: The sensor's control circuitry serves as its brain. It handles everything, from sending out the waves to deciphering the echoes that come back.

Housing: The sensor's body, similar to a shell, which keeps all of the parts safe and in place.

Emitter/Receiver Configuration: The sensor may contain one or two transducers. One for sending and receiving, or distinct for each.

Trigger and Echo Pins: These are similar to the sensor's hands. The trigger pin instructs the sensor when to send out waves, while the echo pin indicates when the waves return and how long they took.

Fig 3.12 Ultrasonic Sensor

8. Pi 4 camera:

The Raspberry Pi 4 camera module is a versatile device that allows users to take high-quality photos and films directly from their Raspberry Pi 4 single-board computer. Here are some important features and details about the Raspberry Pi 4 camera module:

1. Camera Modules: The Raspberry Pi 4 may use two official camera modules: Raspberry Pi Camera Module V. This module has a fixed-focus lens and can take still photographs up to 5 megapixels (2592x1944 pixels). It connects to the Raspberry Pi using a ribbon cable and the CSI (Camera Serial Interface) connector.

-Raspberry Pi Camera Module V2: The V2 camera module improves performance and image quality over its predecessor. It has a Sony IMX219 8-megapixel sensor and can shoot at up to 8 megapixels (3280x2464 pixels). It likewise has a fixed-focus lens and connects to the Raspberry using the CSI connection.

2. Versatile Usage: Raspberry Pi camera modules can be utilized for a wide range of applications, including photography, videography, surveillance, computer vision, robotics, and remote monitoring. They let users capture still photos and videos directly from their Raspberry Pi projects.

3. High-Quality Imaging: Both camera modules provide high-quality imaging capabilities, allowing users to shoot clear, detailed photos and movies. In comparison to the V1 module, the V2 camera module performs better in low light and reproduces colors more accurately.

4. Software Support: The Raspberry Pi camera modules are compatible with the standard Raspberry Pi OS (previously Raspbian), as well as a variety of third-party software libraries and apps. Users can easily access and manage the camera modules via Python scripts or command-line tools, allowing them to collect photographs, record movies, change camera settings, and execute image processing operations.

5. Accessories and Add-ons: Camera covers, lens attachments, infrared (IR) filters, and camera mounts are among the accessories and add-ons available to users for their Raspberry Pi camera projects. The accessories These accessories enhance the functionality and customization choices for various project requirements.

6. Community Projects and Tutorials: The Raspberry Pi community provides a multitude of resources, projects, and tutorials for camera usage on the Raspberry Pi. Users may find step-by-step instructions, example code, and project ideas for integrating camera modules into Raspberry Pi projects, making it simple to get started with camera-based applications.

The Raspberry Pi 4 camera modules provide consumers an economical and convenient way to capture photographs and movies directly from their Raspberry Pi projects. These cameras offer high-quality imaging, flexibility, and significant software support, making them suitable for a variety of applications in both hobbyist and professional settings.

\

Fig 3.13 Raspberry Pi Camera

8. LED (Light Emitting Diode)

LEDs (Light-Emitting Diodes) are widely employed as indicators in robotics, providing visual input on the condition of various components or functions. Here's how LEDs are generally utilized as indicators in robots:

1.Indicators: These LEDs can be used to show which way the robot intends to turn. When the robot prepares to turn right, the right indicator LED will illuminate. Similarly, when the robot prepares to turn left, the left indicator LED illuminates. This offers visible input to nearby people or vehicles, improving safety and communication in traffic.

2. Theft Indicator: This LED can be used as a theft deterrent or alarm indicator. When activated, such as when the robot senses unauthorized movement or tampering, the theft indication LED flashes or remains lit, indicating a possible theft or security violation.

2. Theft Indicator: This LED can be used as a theft deterrent or alarm indicator. When activated, such as when the robot senses unauthorized movement or tampering, the theft indication LED flashes or remains lit, indicating a possible theft or security violation.

3.Object Detected Indicator: This LED lights up when the robot detects an object or barrier in its route. When the robot's sensors identify an object within a predetermined range, the object detected indicator LED turns on. Depending on the proximity and urgency of the identified object, the LED can blink quickly or remain lit continuously to alert the robot's operator to the obstacle. Integrating this indication improves the robot's safety by notifying users of probable collisions or impediments in its environment, allowing for prompt intervention or corrective action.

Fig 3.14 LED

3.2.9 Design and Simulation results

Discover the possibilities of CATIA, a leading 3D CAD software known for its comprehensive tools for engineers and designers. CATIA enables users to accurately and efficiently conceptualize, analyze, and implement complex product designs. CATIA V8, the latest generation, improves the CAD experience with sophisticated features and additions. One major addition is the Model Based Definition (MBD) feature, which simplifies product definition processes by utilizing 3D models rather than traditional 2D drawings. This novel strategy not only lowers errors, but also promotes effective communication and collaboration among team members. CATIA V8 also has a sophisticated array of generative design tools, which use artificial intelligence to explore new design possibilities. Engineers and designers can use this technology to maximize product performance. Engineers and designers can use this technology to improve product performance, reduce material usage, and speed up the innovation cycle.

Whether creating intricate components or complicated assemblies, CATIA V8 allows users to fulfill their design concepts with unrivaled precision and inventiveness. CATIA V8 represents the future of CAD, where innovation knows no bounds.

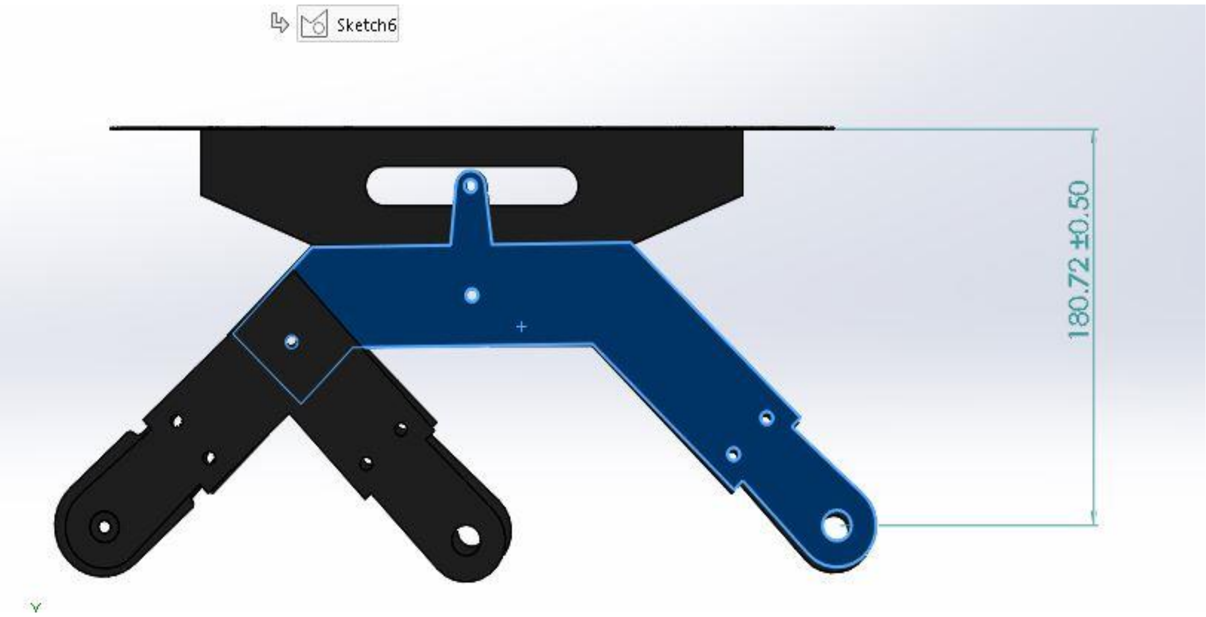


Fig :3.16: Design and Simulation results

CHAPTER 4

Conclusion

The collaborative efforts of a diverse team are crucial in successfully building a food delivery robot. From the initial stages of conceptualization and planning to the integration of mechanical, electronic, and software components, each team member plays a vital role. The iterative process of testing and quality assurance ensures that the robot meets the required standards of functionality, reliability, and safety.

The software development phase was equally critical, with programming and software development creating the intelligence that powers the robot. Control algorithms, navigation systems, and user interfaces were developed to make the robot operate efficiently. The iterative process of testing and quality assurance was systematically applied to verify that the robot met stringent standards of functionality, reliability, and safety. This phase involved continuous refinements and modifications to address any identified issues.

The heart of the project lies in the integration of mechanical, electronic, and software components. designing the physical structure of the robot, ensuring durability, optimal weight distribution, and the successful implementation of a delivery mechanism. Simultaneously, the electrical and hardware group worked on the electronic components, ranging from Raspberry Pi to sensors, actuators, and communication modules. The close coordination among these teams was imperative to ensure the seamless compatibility of the device hardware components.

As technology continues to advance, the innovative solutions developed by such collaborative teams contribute not only to robotics but also to automation and delivery services. The journey from conceptualization to deployment underscores the importance of teamwork, adaptability, and continuous improvement in engineering projects

CHAPTER 5

Future Scope

Increased Autonomy and Navigation Capabilities: Future delivery robots are likely to exhibit enhanced autonomy and navigation capabilities. Advanced artificial intelligence (AI) and machine learning algorithms will enable them to navigate complex environments, including crowded urban areas and dynamic pedestrian traffic.

Integration of Advanced Sensors: Delivery robots will incorporate more advanced sensors and perception systems. This includes improved computer vision, LiDAR, and other sensor technologies, allowing robots to better understand and interact with their surroundings. This will contribute to safer and more efficient navigation.

Collaboration with Autonomous Vehicles: Collaboration between delivery robots and autonomous vehicles is anticipated. Seamless integration between robots and delivery vehicles can optimize logistics operations, especially in urban areas, where robots can assist in the efficient distribution of goods.

Expansion to Various Industries: The use of delivery robots is likely to expand beyond the food and retail sectors. Industries such as healthcare, pharmaceuticals, and logistics may adopt delivery robots to transport medical supplies, prescriptions, and various goods within controlled environments like hospitals and warehouses.

Advanced Human-Robot Interaction: Future delivery robots may feature advanced human-robot interaction capabilities. This includes improved communication interfaces, voice recognition, and even the ability to interact socially with customers or recipients, enhancing user experience.

Integration with Smart Cities: Delivery robots are likely to become integral components of smart city initiatives. They can contribute to the optimization of urban logistics, reducing traffic congestion and emissions by providing efficient and environmentally friendly delivery solutions.

Delivery robots are getting even better in the future! They will be able to navigate through busy cities more safely using improved sensors and smart technology. They'll also use super-fast 5G internet, making them work even more efficiently. These robots won't just be delivering food – they'll help with healthcare, delivering important things in hospitals. The way they talk to people will be friendlier, and they'll be more careful about the environment by using eco-friendly materials and energy-saving parts. In smart cities, these robots will be like helpful assistants, making deliveries without adding to traffic problems. They'll be able to change their parts for different jobs, and there will be rules to make sure they're safe and ethical. The exciting part is that these robots will not only make deliveries easier but also become important in many different fields, making our lives more convenient with the help of technology.

5.1 References

[1]. Saleh Ahmad, Hongwei Zhang, and Guangjun Liu, Senior Member, IEEE. “Multiple Working Mode Control of Door-Opening with a Mobile Modular and Reconfigurable Robot” IEEE/ASME Transactions on Mechatronics (Volume: 18, Issue: 3, June 2013)

[2]. S ‘A Layered Architecture for Office Delivery’ Robots Reid Simmons Richard Goodwin, Karen Zita Haigh, Sven Koenig, Joseph O'Sullivan.

[3]. V.M. BUDANOV and Yeo A. DEVYANIN, ‘The Motions of Wheeled Robots’, IEEE Control Science Direct 14 March 2002.

[4]. J. Evans, B. Krishnamurthy, B. Barrows, T. Skewis, and V. Lumelsky, ‘‘Handling real-world motion planning: A hospital transport robot,’’ IEEE Control Syst

[5]. M. H. A. Hamid, A. H. Adom, N. A. Rahim, and M. H. F. Rahiman, “Guidance of a mobile robot using the Global Positioning System (GPS) and an obstacle avoidance system employing a commanded loop daisy-chaining application approach.” 2015

[6]. A low-cost touchscreen operant chamber using a Raspberry Pi, Behavior Research Methods Proceedings of the Third International Conference on Smart Systems and Inventive Technology, 2015

[7]. Gehlod, Lalit, Vaibhav Jain, Mala Dutta, and D. K. Lal, authored the paper titled "A Grid-Based Robot Navigation by Using Priority Algorithm," published in the International Journal of Advanced Research in Computer and Communication Engineering, Volume 2, August 2013.

[8]. Sakorn Mekruksavanich, “The Smart Shopping Basket Based on IoT Applications”, Software Engineering and Service Science (ICSESS) 2019 10th International Conference, 2019.

[9]. T. K. Das, A. K. Tripathy, and K. Srinivasan, "A Smart Trolley for Smart Shopping," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), 2020.

[10]. “A real Time Object Recognition and Counting System for Smart Industrial Camera Sensor” IEEE SENSORS JOURNAL, NOVEMBER 2017.

[11]. Car Parking Indicator System using Arduino Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021) IEEE Xplore.

[12]. Multiple Working Mode Control of Door-Opening with a Mobile Modular and Reconfigurable Robot This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, except pagination. IEEE/ASME TRANSACTIONS ON MECHATRONICS

[13]. N. Koyachi, H. Adachi, M. Izumi, and T. Hirose, “Control of walk and manipulation by a hexapod with integrated limb mechanism: Melmantis 1,” in Proceedings 2002 IEEE International Conference on Robotics and Automation.

[14]. Object Detection and separation Using Raspberry PI Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE Xplore Compliant

[15]. Multiple Working Mode Control of Door-Opening with a Mobile Modular and Reconfigurable Robot This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, except pagination. IEEE/ASME TRANSACTIONS ON MECHATRONICS

6. Annexures

1.Automation Code

```
import serial
from decimal import *
import RPi.GPIO as GPIO
from time import sleep
import os
import time
import pyrebase

m1pin1=40;
m1pin2=38;
m2pin1=36;
m2pin2=35;
m3pin1=33;
m3pin2=31;
m4pin1=29;
m4pin2=32;
m5pin1=16;
m5pin2=18;
m6pin1=11;
m6pin2=13;

GPIO.setmode(GPIO.BOARD)
#wheel pin
GPIO.setup(m1pin1,GPIO.OUT)
GPIO.setup(m1pin2,GPIO.OUT)
GPIO.setup(m2pin1,GPIO.OUT)
GPIO.setup(m2pin2,GPIO.OUT)
GPIO.setup(m3pin1,GPIO.OUT)
GPIO.setup(m3pin2,GPIO.OUT)
GPIO.setup(m4pin1,GPIO.OUT)
GPIO.setup(m4pin2,GPIO.OUT)
GPIO.setup(m5pin1,GPIO.OUT)
GPIO.setup(m5pin2,GPIO.OUT)
GPIO.setup(m6pin1,GPIO.OUT)
GPIO.setup(m6pin2,GPIO.OUT)

#from picamera import PiCamera
#camera = PiCamera()
#camera.resolution = (1920, 1080)
#camera.start_preview()
#print('Hello')

import pyrebase
config = {
  "apiKey": "5VjeYDr9BRkfkZg8b0JJ3P4sdL0JZtqsBmpfl2AU",
```

```
  "authDomain": "logistics-and-delivery-robot-default-rtdb.firebaseio.com",

  "databaseURL": "https://logistics-and-delivery-robot-default-rtdb.firebaseio.com",

  "storageBucket": "logistics-and-delivery-robot"
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()
print(db)


def find(str, ch):
    for i, ltr in enumerate(str):
        if ltr == ch:
            yield i

def gpsread():
        ser = serial.Serial('/dev/ttyUSB0', 9600,timeout=0.5)
        cd=1
        while cd <= 1:
            ck=0
            fd=""
            while ck <= 1:
                rcv = ser.readline(50)
                rcv = str(rcv)
                #print(rcv)
                fd=fd+rcv
                ck=ck+1

            if '$GPRMC' in fd:
                ps=fd.find('$GPRMC')
                dif=len(fd)-ps
                if dif > 50:
                    data=fd[ps:(ps+50)]
                    #print data
                    ds=data.find('A')
                    if ds > 0 and ds < 20:
                        p=list(find(data, ","))
                        lat=data[(p[2]+1):p[3]]
                        lon=data[(p[4]+1):p[5]]

                        s1=lat[2:len(lat)]
                        s1=Decimal(s1)
                        s1=s1/60
                        s11=int(lat[0:2])
                        s1=s11+s1
```

```
            s2=lon[3:len(lon)]
            s2=Decimal(s2)
            s2=s2/60

            s22=int(lon[0:3])
            s2=s22+s2
            cd=cd+1
            #print s1
            #print s2
            s1=round(s1,6)
            s2=round(s2,6)
            #gps=str(s1)+'  '+str(s2)

            return s1,s2




def forward():
    GPIO.output(m1pin1,1)
    GPIO.output(m1pin2,0)
    GPIO.output(m2pin1,1)
    GPIO.output(m2pin2,0)
    GPIO.output(m3pin1,1)
    GPIO.output(m3pin2,0)
    GPIO.output(m4pin1,1)
    GPIO.output(m4pin2,0)
    GPIO.output(m5pin1,1)
    GPIO.output(m5pin2,0)
    GPIO.output(m6pin1,1)
    GPIO.output(m6pin2,0)
    print('Forward')
    time.sleep(5)
    GPIO.output(m1pin1,0)
    GPIO.output(m1pin2,0)
    GPIO.output(m2pin1,0)
    GPIO.output(m2pin2,0)
    GPIO.output(m3pin1,0)
    GPIO.output(m3pin2,0)
    GPIO.output(m4pin1,0)
    GPIO.output(m4pin2,0)
    GPIO.output(m5pin1,0)
    GPIO.output(m5pin2,0)
    GPIO.output(m6pin1,0)
    GPIO.output(m6pin2,0)
    time.sleep(2)
```

```
def reverse():
    GPIO.output(m1pin1,0)
    GPIO.output(m1pin2,1)
    GPIO.output(m2pin1,0)
    GPIO.output(m2pin2,1)
    GPIO.output(m3pin1,0)
    GPIO.output(m3pin2,1)
    GPIO.output(m4pin1,0)
    GPIO.output(m4pin2,1)
    GPIO.output(m5pin1,0)
    GPIO.output(m5pin2,1)
    GPIO.output(m6pin1,0)
    GPIO.output(m6pin2,1)
    print('Reverse')
    time.sleep(5)
    GPIO.output(m1pin1,0)
    GPIO.output(m1pin2,0)
    GPIO.output(m2pin1,0)
    GPIO.output(m2pin2,0)
    GPIO.output(m3pin1,0)
    GPIO.output(m3pin2,0)
    GPIO.output(m4pin1,0)
    GPIO.output(m4pin2,0)
    GPIO.output(m5pin1,0)
    GPIO.output(m5pin2,0)
    GPIO.output(m6pin1,0)
    GPIO.output(m6pin2,0)
    time.sleep(2)

def stop():
    GPIO.output(m1pin1,0)
    GPIO.output(m1pin2,0)
    GPIO.output(m2pin1,0)
    GPIO.output(m2pin2,0)
    GPIO.output(m3pin1,0)
    GPIO.output(m3pin2,0)
    GPIO.output(m4pin1,0)
    GPIO.output(m4pin2,0)
    GPIO.output(m5pin1,0)
    GPIO.output(m5pin2,0)
    GPIO.output(m6pin1,0)
    GPIO.output(m6pin2,0)
    print('stop')
    time.sleep(0.5)
```

```
def right():
    print('right')
    GPIO.output(m2pin1,0)
    GPIO.output(m2pin2,1)
    GPIO.output(m3pin1,0)
    GPIO.output(m3pin2,1)
    GPIO.output(m6pin1,0)
    GPIO.output(m6pin2,1)

    GPIO.output(m1pin1,1)
    GPIO.output(m1pin2,0)
    GPIO.output(m4pin1,1)
    GPIO.output(m4pin2,0)
    GPIO.output(m5pin1,1)
    GPIO.output(m5pin2,0)

    time.sleep(5)
    GPIO.output(m1pin1,0)
    GPIO.output(m1pin2,0)
    GPIO.output(m2pin1,0)
    GPIO.output(m2pin2,0)
    GPIO.output(m3pin1,0)
    GPIO.output(m3pin2,0)
    GPIO.output(m4pin1,0)
    GPIO.output(m4pin2,0)
    GPIO.output(m5pin1,0)
    GPIO.output(m5pin2,0)
    GPIO.output(m6pin1,0)
    GPIO.output(m6pin2,0)
    time.sleep(2)

def left():
    print('left')
    GPIO.output(m1pin1,0)
    GPIO.output(m1pin2,1)
    GPIO.output(m4pin1,0)
    GPIO.output(m4pin2,1)
    GPIO.output(m5pin1,0)
    GPIO.output(m5pin2,1)

    GPIO.output(m2pin1,1)
    GPIO.output(m2pin2,0)
    GPIO.output(m3pin1,1)
    GPIO.output(m3pin2,0)
    GPIO.output(m6pin1,1)
    GPIO.output(m6pin2,0)
```

```
    time.sleep(5)
    GPIO.output(m1pin1,0)
    GPIO.output(m1pin2,0)
    GPIO.output(m2pin1,0)
    GPIO.output(m2pin2,0)
    GPIO.output(m3pin1,0)
    GPIO.output(m3pin2,0)
    GPIO.output(m4pin1,0)
    GPIO.output(m4pin2,0)
    GPIO.output(m5pin1,0)
    GPIO.output(m5pin2,0)
    GPIO.output(m6pin1,0)
    GPIO.output(m6pin2,0)
    time.sleep(2)


try:
    while True:
#        s1,s2= gpsread()
#        print(s1)
#        print(s2)
#        #print(type(float(s1)))
#        db.update({"Lat":float(s1),"Long":float(s2)})
        print("Reading Cloud")
        value = db.child("STATUS").get()
        status=value.val()
        print(status)
        if status=="1":
            forward()
        if status=="2":
            reverse()
        if status=="3":
            right()
        if status=="4":
            left()
        if status=="5":
            stop()




except KeyboardInterrupt:
    print('Stopped by user')
    GPIO.cleanup()
```

2.RFID Door Lock System

```
 #include <Wire.h>
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
#define LED_G 4 // Green LED pin
#define LED_R 5 // Red LED pin
#define BUZZER 2 // Buzzer pin
#define lock 3 // Door lock pin
#define Btn 6 // Manual access grant button pin

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup () {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init ();

  Pin Mode(LED_G, OUTPUT);
  Pin Mode(LED_R, OUTPUT);
  Pin Mode(BUZZER, OUTPUT);
  Pin Mode(lock, OUTPUT);
  Pin Mode(Btn, INPUT);
}

void loop () {
  // Check if the manual access grant button is pressed
  if (digital Read(Btn) == HIGH) {
    grant Access();
  }

  // Check for new RFID cards
  if (mfrc522.PICC_IsNewCardPresent () && mfrc522.PICC_ReadCardSerial ()) {
    String content = "";
    for (byte i = 0; i < mfrc522.uid.size; i++) {
      content.concat(String (mfrc522.uid.uidByte[i] < 0x10? " 0": " "));
      content.concat(String (mfrc522.uid.uidByte[i], HEX));
    }
    content.toUpperCase();
```

```
    // Compare scanned card's UID with predefined UID
    if (content.substring(1) == "83 23 38 BB") {
      grantAccess();
    } else {
      accessDenied();
    }
  }
}

// Function to grant access
void grantAccess() {
  Serial.println("Access Granted");

  digitalWrite(LED_G, HIGH);
  tone (BUZZER, 2000);
  delay (100);
  noTone(BUZZER);
  delay (50);
  tone (BUZZER, 2000);
  delay (100);
  noTone(BUZZER);

  digitalWrite(lock, HIGH); // Activate door lock
  delay (3000); // Keep the lock activated for 3 seconds
  digitalWrite(lock, LOW); // Deactivate door lock

  digitalWrite(LED_G, LOW);
  tone (BUZZER, 2000);
  delay (100);
  noTone(BUZZER);
  delay (50);
}

// Function to indicate access denied
void accessDenied() {
  Serial.println("Access Denied");

  for (int i = 0; i < 3; i++) {
    digitalWrite(LED_R, HIGH);
    tone (BUZZER, 1500);
    delay (500);
    digitalWrite(LED_R, LOW);
    noTone(BUZZER);
    delay (100);
  }
}
```

3.ESpeak

To install and configure espeak for text-to-speech functionality on Raspberry Pi

sudo apt //get install espeak - to install espeak
configure the pi for audio - put it to audio jack
to increase or reduce the volume use - alsamixer
to change the the voice u can use use -vafm3 for male and -vaff4
to use espeak just type - espeak " welcome to dsu"
if you want to read the file type - espeak -f a.text
Use sudo apt-get install espeak to get it installed.
Put the Pi on the audio jack after setting it up for audio.
To raise or lower the sound used - Alsamixer
In order to communicate using Espeak, simply enter espeak "welcome to DSU."
To speak the file type, type espeak -f a.text.

"sudo apt-get install espeak"
"alsamixer"
espeak "welcome to Dayananda sagar University"

4.Camera integration to Pi 4

To connect the MI camera to the Raspberry Pi 4 while setting up a camera streaming service using Motion,

$sudo apt-get install motion // to install the Motion software:
$Sudo nano /etc/motion/motion.conf // to access the Motion configuration file
$sudo service motion restart // After specifying the motion parameters, restart the Motion $service using the following command
$sudo motion // Launch Camera Service
http://<Raspberry_Pi_IP_Address>:<Port_Number>// Access Camera Feed