Mini Project Report
On

"AUTONOMOUS DELIVERY ROBOT WITH PASSWORD AUTHENTICATION"

A Mini project report submitted in the partial fulfillment of the Academic Requirements for the award of

Bachelor of Technology
In
Electronics & Communication Engineering
Submitted by

U. Ganesh	**235U5A0412**
M. Rashmi Priya	**235U5A0410**
P. Nithin Reddy	**225U1A04A1**
P. Anil	**225U1A0495**

Under the guidance of
Mr. MD. Gandhi Babu,
Asst. Professor, Dept of E.C.E

Department of Electronics and Communication Engineering

AVN INSTITUTE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E. & Govt. of Telangana & Affiliated to JNTU Hyderabad)

Accredited by NAAC & NBA

PATEL GUDA, IBRAHIMPATANAM (M), R.R-Dist-501510-TS

2025-2026

CERTIFICATE

This is to certify that the dissertation entitled **"AUTONOMOUS DELIVERY ROBOT WITH PASSWORD AUTHENTICATION"** is a bonafide work done and submitted by

U. Ganesh	**235U5A0412**
M. Rashmi Priya	**235U5A0410**
P. Nithin Reddy	**225U1A04A1**
P. Anil	**225U1A0495**

In partial fulfillment of requirement for the award of **Bachelor of Technology** degree in **Electronics and Communication Engineering** at AVN Institute of Engineering & Technology, Hyderabad.

Certified further that to the best of my knowledge the work in this dissertation has not been submitted to any other university or institute for the award of any degree.

Co-Ordinator
Mrs. B. Mamatha
Asst. Professor
Dept. of ECE.

Head of the Department
Dr. Suresh Narsimhan
Professor & HOD
Dept of E.C.E.

Internal guide
Mr. MD. Gandhi Babu
Asst. Professor
Dept of E.C.E.

External Examiner

DECLARATION

We, here by declare that the work which is being presented in this dissertation entitled **"AUTONOMOUS DELIVERY ROBOT WITH PASSWORD AUTHENTICATION"**, submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Electronics and Communication Engineering**, JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD is a bonafide record of AVN INSTITUTE OF ENGINEERING AND TECHNOLOGY and has not been submitted to any other course or university for award of any degree.

U. Ganesh	**235U5A0412**
M. Rashmi Priya	**235U5A0410**
P. Nithin Reddy	**225U1A04A1**
P. Anil	**225U1A0495**

AVN INSTITUTE OF ENGINEERING & TECHNOLOGY
Accredited by NAAC & NBA

ACKNOWLEDGEMENT

We are greatly thankful to the following people for their contribution, advice and assistance towards the development of this project and also in the completion of the report entitled **"AUTONOMOUS DELIVERY ROBOT WITH PASSWORD AUTHENTICATION"**

We express our sincere thanks to my Co-ordinator **Mrs. B. Mamatha,** Assistant Professor for the guidance, she showed us and for her sincere effort in building the project right direction.

We express our sincere thanks to My Guide **Mr. MD. Gandhi Babu,** Assistant Professor for the guidance, he showed me and for his sincere effort in building the project in right direction.

We are indebted to **Dr. N. Suresh,** Head of the Department of ECE for his valuable suggestion and for the motivation he provided. My sincere thanks to him for sparing his valuable time and constant encouragement throughout the project.

We are highly indebted to Principal, **Dr. P. Naageswara Reddy,** for the support who has not only shown utmost patience, but fertile in suggestions, vigilant in directions of error and who have been infinitely helpful.

U. Ganesh	**235U5A0412**
M. Rashmi Priya	**235U5A0410**
P. Nithin Reddy	**225U1A04A1**
P. Anil	**225U1A0495**

ABSTRACT

An autonomous delivery robot with password authentication integrates advanced robotics with secure access control to ensure safe and efficient deliveries. This system typically employs AI-driven navigation, obstacle detection, and real-time tracking to facilitate autonomous movement, while password authentication ensures that only authorized recipients can access the delivered items. The robot operates using GPS and sensor-based technology to map routes, avoid obstacles, and reach destinations autonomously. It is equipped with a secure storage compartment that requires password authentication for retrieval. Users receive a unique access code upon delivery, which must be entered on the robot's interface or a mobile application to unlock the compartment.

Such a system enhances security by preventing unauthorized access and theft, making it ideal for applications like last-mile deliveries, secure document transportation, or food delivery services. The use of encryption and multi-factor authentication can further bolster security, ensuring that deliveries remain protected and accessible only to intended recipients. Moreover, these robots can be integrated with cloud-based systems for seamless coordination. Fleet management platforms enable operators to track and manage multiple units simultaneously, ensuring deliveries are completed on schedule. Additionally, they can be programmed to return to designated charging stations autonomously, keeping operations sustainable and uninterrupted.

TABLE OF CONTENTS

Contents **Page No**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The project is aimed at evaluating the performance of an operating system on an embedded system. Before delving into its implementation, an introduction is needed to the parts involved in the project. The whole report is centered around the field of embedded systems and the use of Surveillance to run applications on them. Hence an introduction to Embedded Systems and using Surveillance as an OS in them is provided.

1.1 Embedded Systems

An embedded system is a special purpose computer system that is designed to perform very small sets of designated activities. Embedded systems date back as early as the late 1960s where they used to control electromechanical telephone switches. The first recognizable embedded system was the Apollo Guidance Computer developed by Charles Draper and his team. Later they found their way into the military, medical sciences and the aerospace and automobile industries.

These were used to perform dedicated functions within larger systems. Unlike general-purpose computers, embedded systems integrate hardware and software to execute specific tasks efficiently. They are commonly found in automobiles, medical devices, industrial machines, and consumer electronics. Embedded systems typically use microcontrollers or microprocessors and operate with minimal user intervention. They can be real-time systems, ensuring precise execution within strict time constraints. Their applications range from automated control systems to smart appliances, making them essential in modern technology

Today they are widely used to serve various purposes like:

• Network equipment such as firewall, router, switch, and so on.

• Consumer equipment such as MP3 players, cell phones, PDAs, digital cameras, camcorders, home entertainment systems and so on.

• Household appliances such as microwaves, washing machines, televisions and so on.

• Mission-critical systems such as satellites and flight control.

The key factors that differentiate an embedded system from a desktop computer:

• They are cost sensitive.

• Most embedded systems have real time constraints.

• There are multitudes of CPU architectures such as ARM, MIPS, PowerPC that are used in embedded systems. Application-specific processors are employed in embedded systems.

• Embedded Systems have and require very few resources in terms of ROM or other I/O devices as compared to a desktop computer.

1.1.1 Types of Setup

Embedded systems typically consist of a host and a target, where the host is usually a personal computer used for development, and the target is the actual hardware executing embedded applications. One common setup is the Linked Setup, in which the host and target are permanently connected via a physical cable, such as a serial cable or Ethernet link.

This setup eliminates the need for transferring physical storage devices between the two systems. The host is responsible for running the cross-platform development environment, while the target contains essential components like a bootloader, functional kernel, and minimal root filesystem to execute embedded applications efficient.

Removable Storage Setup:

In the removable setup, there are no direct physical links between the host and the target. Instead, a storage device is written by the host, is then transferred into the target, and is used to boot the device. The host contains the cross-platform development environment. The target, however, contains only a minimal bootloader. The rest of the components are stored on a removable storage media, such as a CompactFlash IDE device, MMC Card, or any other type of removable storage device.

Standalone Setup:

The target is a self-contained development system and includes all the required software to boot, operate, and develop additional software. In essence, this setup is similar to an actual workstation, except the underlying hardware is not a conventional workstation but rather the embedded system itself. This one does not require any cross-platform development environment, since all development tools run in their native environments. Furthermore, it does not require any transfer between the target and the host, because all the required storage is local to the target.

1.2 Operating Systems

In an embedded system, when there is only a single task that is to be performed, then only a binary is to loaded into the target controller and is to be executed. However, when there are multiple tasks to be executed or multiple events to be handled, then there has to be a program that handles and prioritizes these events. This program is the Operating System (OS), which one is very familiar with, in desktop PCs.

Various Operating Systems:

Embedded Operating Systems are classified into two categories:

1. Real-time Operating Systems (RTOS) :

Real Time Operating Systems are those which guarantee responses to each event within a defined amount of time. This type of operating system is mainly used by time-critical applications such as measurement and control systems. Some commonly used RTOS for embedded systems are: VxWorks, OS-9, Symbian, RTLinux.

2. Non-Real-time Operating Systems:

Non-Real Time Operating Systems do not guarantee defined response times. These systems are mostly used if multiple applications are needed. Windows CE and PalmOS are examples for such embedded operating systems.

Why Linux?

There are a wide range of motivations for choosing Linux over a traditional embedded OS.

The following are the criteria due to which Linux is preferred:

1. Quality and Reliability of Code:

Quality and reliability are subjective measures of the level of confidence in the code that comprises software such as the kernel and the applications that are provided by distributions. Some properties that professional programmers expect from a “quality” code are modularity and structure, readability, extensibility and configurability. “Reliable” code should have features like predictability, error recovery and longevity.

2. Availability of Code:

Code availability relates to the fact that the Linux source code and all build tools are available without any access restrictions. The most important Linux components, including the kernel itself, are distributed under the GNU General Public License (GPL).Access to these components' source code is therefore compulsory (at least to those users who have purchased any system running GPL-based software, and they have the right to redistribute once they obtain the source in any case). Code availability has implications for standardization and commoditization of components, too. Since it is possible to build Linux systems based entirely upon software for which source is available, there is a lot to be gained from adopting standardized embedded software platforms.

3. Hardware Support:

Broad hardware support means that Linux supports different types of hardware platforms and devices. Although a number of vendors still do not provide Linux drivers, considerable progress has been made and more is expected. Because a large number of drivers are maintained by the Linux community itself, you can confidently use hardware components without fear that the vendor may one day discontinue driver support for that product line. Linux also provides support for dozens of hardware architectures. No other OS provides this level of portability.

1.3 Surveillance:

Surveillance systems can be software-based or hardware-based, and many surveillance applications do rely on an operating system (OS) for functionality. Surveillance software, such as video monitoring programs, security management tools, and network surveillance applications, typically run on operating systems like Windows, Linux, or specialized security-focused OS.

Some surveillance systems are embedded within devices, meaning they operate on firmware or embedded OS rather than traditional operating systems. For example, network video recorders (NVRs) and digital video recorders (DVRs) often use custom embedded OS optimized for security and video processing.

CHAPTER 2

LITERATURE SURVEY

With embedded systems fast expanding its reach, subject matter related to this field is available in abundance. While working on this project we have studied matter from various sources such as books, online articles and reference manuals. The knowledge gained from this activity has been of great help to us in understanding the basic concepts related to our project and has ignited further interest in this topic.

Research highlights the vulnerabilities of traditional password-based systems and proposes multi-factor authentication schemes that incorporate blockchain technology to enhance security and prevent cyber threats. AI-driven robotic delivery systems have been developed to improve real-time decision-making and adaptability in dynamic urban environments, utilizing encrypted communication protocols between the robot and the recipient to ensure secure authentication. Here are some authors who have contributed to **Autonomous Delivery Robot With Password Authentication.**

Shalom Tirkey & Thotakura Ayyappa – Their work on the design and fabrication of self-driving delivery robots explores authentication mechanisms like OTP-based access for secure package retrieval.

Yang Yang, Aryan Mohammadi Pasikhani, Prosanta Gope, & Biplab Sikdar – They proposed a privacy-preserving multi-factor authentication scheme for secure automated delivery systems, addressing security threats like impersonation and replay attacks.

Elin Alverhed, Simon Hellgren, Hanna Isaksson, Lisa Olsson, Hanna Palmqvist, & Jonas Flodén – Their literature review investigates how self-driving autonomous delivery robots impact last-mile deliveries and contribute to logistics and transport industries.

Alexander Buchegger et.al,2018. To allow autonomous transport vehicles to be used for transportation tasks in large-scale outdoor environments proven approaches from the robotics domain needs to be applied and transferred to these new environments. In this paper, we present an integrated autonomous transport vehicle which addresses these problems and is able to deliver parcels in urban environments such as city centers automatically. The developed transport vehicle is based on a commercial electrical personal vehicle. It was adapted for autonomous control and equipped with improved navigation skills for outdoor environments based on a topological navigation approach. The integrated vehicle was evaluated in realistic delivery use cases where parcels are delivered autonomously to addresses in a larger urban area. The adaptation of well-known algorithms for robot navigation for large-scale urban environments An integration of these algorithms in a commercially available electrical vehicle. The improvement of the robustness of the approach by integrating additional from OpenStreetMap (OSM) an evaluation of the autonomous delivery concept inreal urban environments such as an university campus or a city center.

Aniket Gujarathi et.al, 2019 The field of autonomous robots is growing rapidly in the world, in terms of both the diversity of emerging applications and the levels of interest among traditional players in the automotive, truck, public transportation, industrial, and military communities. Autonomous robotic systems offer the potential for significant enhancements in safety and operational efficiency. Due to the meteoric growth of e- commerce, developing faster, more affordable and sustainable last-mile deliveries become more important. In this paper, Autonomous robot including the cyber physical architecture of the robots as well as the renderings of CAD models are illustrated. Designing new solutions including catadioptric cameras that output panoramic views of the scene, i.e., images with very large fields of view. It describes the problem of state estimation and localization of a robot in detail. In order to navigate can move smoothly only if itis properly localized. An inaccurate localization may cause the robot to vary on the roads or behave erroneously which are serious issues when the robot is completely autonomous.

Murad Mehrab Abrar et.al, 2021 Robots and autonomous vehicles can help to ease the stress on the existing home delivery while reducing the risk of virus transmission by mitigating direct human contact. In this regard, we have developed a cost effective autonomous mobile robot prototype for the purpose of increasing the last mile delivery efficiency as well as ensuring a secure and contactless package delivery. An autonomous mobile robot is a self-driving vehicle that does not require any operation from operator to navigate the robot. The movements and trajectory are predefined before the operation and the robot navigates accordingly. Among various navigation techniques, we have used the Global Positioning System

(GPS) data for autonomous navigation of the robot and the destination is predefined as latitude and longitude points in the program of the robot. The main advantage of using GPS for navigation is that the data received from the GPS are independent of the previous readings; therefore, it is easy to minimize errors. A digital compass measures the heading angle of the robot and helps the robot to find the direction of the trajectory. The robot is equipped with a password protected container which protects the package against theft, damage and unprotected human contact. This password can be sent to the customer by a text message from the service company. Once the robot arrives at its delivery location, the only person who has the password will be able to unlock its delivery.

Fake fingerprint/pin/password generation:

Before describing the most widely used fake fingerprint databases which are publicly available, and in order to help to understand their structure, we believe it is useful to present here a brief summary of the most common techniques used for the generation of gummy fingers. the creation of fake fingers is in almost all cases carried out following one of three procedures depending on the starting point of the manufacturing process: starting from the user's finger this method is also known as "cooperative" and further reading may be found for instance in. In this case, the legitimate user is asked to place his finger on a mouldable and stable material in order to obtain the negative of the fingerprint.

In a posterior step the gummy finger is recovered from the negative mould starting from a latent fingerprint this method is also referred to in many publications as "non-cooperative" and was first introduced in. In this case the first step is to recover a latent fingerprint that the user has unnoticed left behind (e.g., on a cd). The latent fingerprint is lifted using a specialized fingerprint development toolkit and then digitalized with a scanner. The scanned image is then enhanced through image processing and finally printed on a pcb from which the gummy finger is generated

Fake biometrics means by using the real images like iris images captured from a printed paper or fingerprint captured from a dummy finger of human identification characteristics create the fake identities like fingerprint, iris on printed paper. Fake user first captures the original identities of the genuine user and then they make the fake sample for authentication. There is no such technology to provide security for fake users.

2.1 EXISTING SYSTEM

Existing systems for autonomous delivery robots with password authentication focus on enhancing security and efficiency in last-mile delivery. One approach involves privacy-preserving multi-factor authentication, which integrates AI-assisted security measures to protect against impersonation, replay attacks, and identity theft. These systems ensure that only authorized recipients can access delivered packages using secure authentication method. Another system emphasizes versatile delivery robots that incorporate GPS navigation, secure locking mechanisms, and real-time monitoring to optimize delivery routes and enhance security. These robots use encrypted communication protocols to prevent unauthorized access and ensure safe package retrieval. Additionally, some projects, such as the MIT-Mentors autonomous delivery robot, utilize ROS framework and web applications for document delivery in controlled environments like college campuses. These robots autonomously navigate between sender and receiver locations, employing security features such as electromagnet-based locking systems.

PROPOSED METHOD

The proposed method for autonomous delivery robots with password authentication focuses on enhancing security and efficiency in last-mile delivery. This system integrates multi-factor authentication, including one-time passwords (OTP), PIN codes, and biometric verification such as facial recognition and voice authentication, to ensure that only authorized recipients can access delivered packages. To improve security, privacy-preserving authentication schemes are implemented, utilizing blockchain technology to prevent cyber threats like impersonation and replay attacks. Additionally, AI-driven authentication modules enable real-time verification, ensuring seamless interaction between the robot and the recipient. The delivery robot is equipped with encrypted communication protocols, allowing secure data exchange and preventing unauthorized access to sensitive information

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1 BLOCK DIAGRAM

Fig 3.1 BLOCK DIAGRAM

3.2 ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.It is used in robotics, home automation, IoT projects, industrial automation, and education.

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.
There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50

- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

Arduino plays a vital role in your project by enabling sensor integration, automation, and control. It can be used to interface with components like LCD displays, keypads, motors, and communication modules, enhancing functionality. In an autonomous delivery robot, Arduino can manage navigation, obstacle detection, and package verification by processing sensor data and controlling actuators. Additionally, it facilitates wireless communication, allowing remote monitoring and control via Wi-Fi or Bluetooth. If your project involves security systems, Arduino can handle password authentication using a 4×4 keypad and display status updates on an LCD screen. It is also widely used in IoT applications, home automation, and robotics, making it a versatile tool for various innovations.

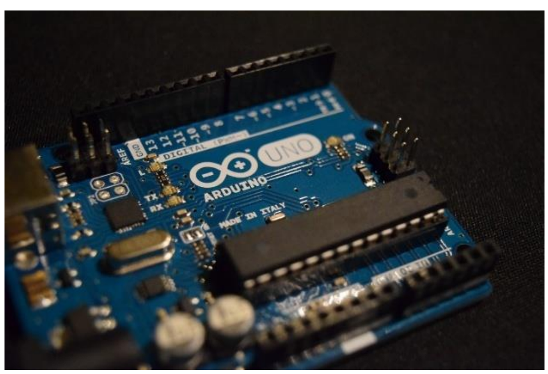


Fig 3.1. ARDUINO

3.2 MOTOR DRIVER(L298N)

The L298N is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage , high current dual full-bridge driver de-signed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals .The emitters of the lower transistors of each bridge are connected together rand the corresponding external terminal can be used for the connection of an external sensing resistor. An additional Supply input is provided so that the logic works at a lower voltage. L298N module is a high voltage, high current dual full-bridge motor driver module for controlling DC motor and stepper motor. It can control both the speed and rotation direction of two DC motor. This module consists of an L298 dual-channel H-Bridge motor driver IC. This module uses two techniques for the control speed and rotation direction of the DC motors. These are PWM – For controlling the speed and H-Bridge – For controlling rotation direction. These modules can control two DC motor or one stepper motor at the same time.

L298 is a high voltage, high current dual full-bridge motor driver IC. It accepts standard TTL logic levels (Control Logic) and controls inductive loads such as relays, solenoids, DC and Stepper motors. This is a 15 pin IC. According to the L298 datasheet, its operating voltage is +5 to +46V, and the maximum current allowed to draw through each output 3A. This IC has two enable inputs, these are provided to enable or disable the device independently of the input signals. A black color heat sink is attached to the L298 IC of the module. A heat sink is a passive heat exchanger that transfers the heat generated by an electronic or a mechanical device to a fluid medium, often air or a liquid coolant.With a maximum motor supply voltage of 46V and a current rating of 2A per channel, the L298N is suitable for robotics, automation, and motorized projects. It includes a 78M05 voltage regulator, which provides a stable 5V output when the input voltage is below 12V. The module also has ENA and ENB pins for enabling motor control and IN1, IN2, IN3, and IN4 pins for setting motor direction.

The H-bridge circuit inside the L298N allows motors to rotate in both directions by switching the polarity of the voltage applied. This makes it ideal for robotic arms, wheeled robots, and motorized platforms. Additionally, the heat sink on the module helps dissipate excess heat, ensuring stable operation.

The L298N motor driver is a crucial component in your project, especially if you're working with autonomous delivery robots or motorized systems. It allows you to control DC motors and stepper motors, managing both speed and direction efficiently.

In your autonomous delivery robot, the L298N can be used to drive the motors responsible for movement and navigation. By utilizing PWM (Pulse Width Modulation), you can adjust the speed of the motors dynamically, ensuring smooth operation. The H-Bridge configuration of the L298N enables bidirectional control, allowing the robot to move forward, backward, and even turn precisely. Additionally, the L298N module can be integrated with Arduino or ESP32, making it easy to program and control via microcontrollers. It supports dual motor control, meaning you can manage two motors simultaneously, which is ideal for robotic platforms. The module also includes a 5V regulator, which can power the microcontroller, reducing the need for additional components.

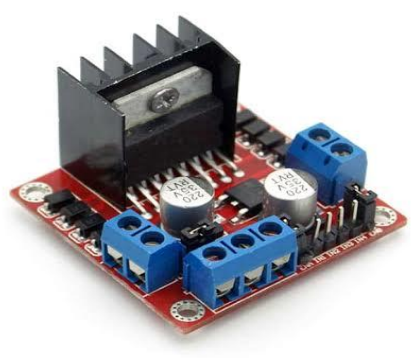


Fig 3.2 MOTOR DRIVE

3.3 ESP32 WEB CAM

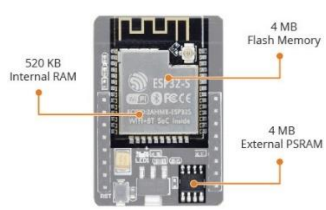
The ESP32-CAM module is a compact and powerful camera module based on the ESP32-S chip. It features an OV2640 camera sensor with a 2MP resolution, making it ideal for applications like video streaming, facial recognition, and IoT-based surveillance. The module supports Wi-Fi and Bluetooth, allowing wireless communication and remote access to captured images or video feeds.

One of its standout features is the microSD card slot, which enables image storage and data logging. Additionally, it comes with 4MB of PSRAM, which enhances its ability to process images efficiently. The ESP32-CAM is widely used in smart home security systems, AI-powered image processing, and embedded vision projects. There are plenty of exciting ways to use the ESP32-CAM in your project! You can explore motion detection systems that trigger alerts or capture images when movement is detected. If you're into robotics, you could build a Mars Rover replica or a hexapod spider bot using the ESP32-CAM. For AI applications, you can integrate gesture-controlled virtual mouse functionality or industrial product counters powered by OpenCV.

For navigation, the camera can work alongside ultrasonic sensors or LiDAR to detect obstacles and plan routes efficiently. Machine learning models like YOLOv3 can be integrated to recognize objects and ensure safe movement. Additionally, the ESP32-CAM can stream live video to a web interface, allowing remote operators to monitor the robot’s journey.

The Memory

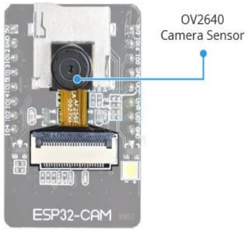
Memory is paramount for complex tasks, so the ESP32-S has a full 520 kilobytes of internal RAM, which resides on the same die as the rest of the chip’s components.



It may be inadequate for RAM-intensive tasks, so ESP32-CAM includes 4 MB of external PSRAM (Pseudo-Static RAM) to expand the memory capacity. This is plenty of RAM, especially for intensive audio or graphics processing.

All these features amount to nothing if you don't have enough storage for your programs and data. The ESP32-S chip shines here as well, as it contains 4 MB of on-chip flash memory.

The Camera
The OV2640 camera sensor on the ESP32-CAM is what sets it apart from other ESP32 development boards and makes it ideal for use in video projects like a video doorbell or nanny cam.



The OV2640 camera has a resolution of 2 megapixels, which translates to a maximum of 1600×1200 pixels, which is sufficient for many surveillance applications.

The Storage
The addition of a microSD card slot on the ESP32-CAM is a nice bonus. This allows for limitless expansion, making it a great little board for data loggers or image capture.

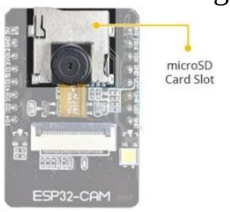


Fig 3.3 ESP32 WEBCAM

3.4 SERVO MOTOR(MG995)

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. The motor moves to a specific position in response to a control signal. Servomotors can be rotary or linear, and can be electric, hydraulic, pneumatic, or steam-powered. The name "servomotor" comes from the Latin words "servus" (slave or servant) and "motor" (mover).

The servo is suited for designing robotic arm in which wear and tear of motor is high. Being metal geared, the servo has long life and can be installed on system like robotic arm were motor work is huge. The servo is also suited to be used in drones and toy planes. Having a satisfying torque which is enough to overcome air resistance and control wings of plane, the servo is preferred in toy planes and drones which need precision control no matter the condition.

The MG995 servo motor is a high-torque, metal gear servo widely used in robotics, automation, and RC applications. It features dual ball bearings for durability and provides precise angular control using pulse width modulation (PWM). Operating at a voltage range of 4.8V to 7.2V, it delivers a torque of 8.5 kg·cm at 4.8V and 10 kg·cm at 6V, making it suitable for applications requiring strong rotational force. The motor has a rotation angle of 180° and operates at speeds of 0.2 sec/60° at 4.8V and 0.16 sec/60° at 6V, ensuring quick response and stable movement. The MG995 servo motor works on PWM signals, where the duty cycle determines the rotation angle. A 0.5ms pulse moves the motor to 0°, a 1.5ms pulse moves it to 90°, and a 2.5ms pulse moves it to 180°, allowing precise control over movement.

It is widely used in robotic arms, RC vehicles, industrial automation, and DIY projects due to its reliability and performance. The servo motor receives signals from the microcontroller and responds with high torque-to-weight ratio, allowing smooth and efficient movement. It functions on pulse-width modulation (PWM) control, enabling stable and adjustable rotation for different locking configurations. Its compact size and energy efficiency make it well-suited for robotic applications, minimizing power consumption while maintaining reliability. When the correct password is entered via the 4×4 keypad, the servo motor activates to unlock the compartment, ensuring secure and authorized access to deliveries. A servo motor is essential in an autonomous delivery robot with password authentication, as it enables precise control over locking and unlocking mechanisms. It operates using a closed-loop feedback system, ensuring accurate angular position control, making it ideal for securing the package compartment. The MG995 servo motor is widely used in robotics, automation, and control systems because of its high torque and precision. The MG995 can be easily interfaced with microcontrollers such as Arduino, where it is commonly used for motion control tasks. Additionally, it finds applications in radio-controlled (RC) vehicles and drones, helping in steering and stabilization. If you are working on a project that demands precise motion control and reliability, integrating the MG995 could significantly improve efficiency and functionality. This servo motor is known for its durability and reliability, making it a great choice for applications requiring precise movement and high torque.

Fig 3.4 SERVO MOTOR

3.5 DC MOTOR

DC motors are electrical machines that convert direct current (DC) electrical energy into mechanical energy. They are widely used in robotics, electric vehicles, industrial machinery, and household devices due to their ability to provide precise speed control.These motors, controlled by the **L298N motor driver module**, enable the robot to move forward, backward, and turn efficiently across different terrains. The system integrates a **4WD car kit**, ensuring stability and traction while transporting packages securely. Once a user enters the correct password via the **4×4 keypad**, the robot unlocks the delivery compartment using a **servomotor**, allowing access to the package. Additionally, **ultrasonic sensors** help detect obstacles, adjusting the motor speed and direction to prevent collisions.

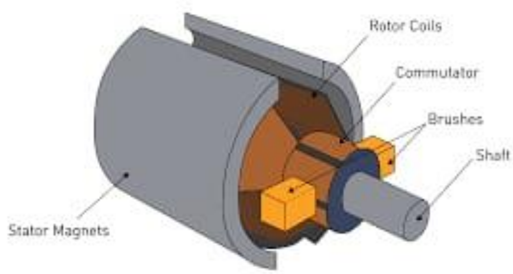


Fig 3.5 DC MOTOR

3.6 BATTERY

A battery is a device that stores and provides electrical energy through chemical reactions. It consists of one or more electrochemical cells, each containing a cathode (positive terminal), an anode (negative terminal), and an electrolyte. In an autonomous delivery robot with password authentication, the battery serves as the primary power source, ensuring uninterrupted operation of all electronic components. Your project utilizes a 7-12V DC LiPo 3S battery, which provides a stable power supply to the Arduino UNO, ESP32-CAM module, motor driver, servomotor, and sensors.

3.7 LCD DISPLAY

An LCD display (Liquid Crystal Display) is a flat-panel screen that uses liquid crystals and a backlight to produce images. It is widely used in electronic devices such as televisions, computer monitors, and embedded systems. In your autonomous delivery robot with password authentication, the 16×2 LCD module serves as a visual interface, displaying important information such as authentication prompts, delivery status, and system alerts. The LCD works by controlling the alignment of liquid crystal molecules between two polarizing filters, allowing light to pass through and form characters or graphics. It is energy-efficient, lightweight, and provides clear readability, making it ideal for real-time data display in robotics applications.

Using an LCD display with an ESP32 offers several advantages, including real-time data visualization, user interaction, and system feedback. LCDs provide a clear and structured way to display sensor readings, system status, or interactive menus, making them ideal for IoT applications, smart home automation, and industrial control systems. The ESP32 supports various LCD interfaces such as SPI, I2C, and RGB, allowing flexibility in design and implementation. Additionally, LCDs enhance usability by providing a graphical interface for users to interact with devices efficiently. However, there are some disadvantages, such as higher power consumption compared to simpler LED indicators, limited resolution in smaller displays, and complex wiring when integrating advanced graphical interfaces.

Fig 3.7 LCD DISPLAY

3.8 4x4 KEYPAD

A 4×4 keypad is an input device consisting of 16 keys arranged in a 4-row by 4-column matrix. It is commonly used in embedded systems and robotics for user input, such as entering passwords or selecting options. In your autonomous delivery robot with password authentication, the 4×4 keypad allows users to input a secure password, which is then processed by the Arduino UNO to unlock the delivery compartment using a servo motor. The keypad is easy to integrate with microcontrollers like Arduino UNO, enabling seamless password authentication. When a user enters the correct password, the system verifies the input and triggers the servo motor to unlock the delivery compartment, ensuring secure access.

Using a 4×4 keypad with an ESP32 offers several benefits. It provides a compact and efficient input method for projects requiring user interaction. The keypad is easy to interface using GPIO pins and can be programmed with libraries like Keypad.h in Arduino or MicroPython. Additionally, it allows for secure password entry, making it ideal for access control systems.

The 4×4 keypad is widely used in security systems, home automation, and robotics. It can be integrated into password-protected door locks, vending machines, and industrial control panels. In ESP32-based projects, it is often used for menu navigation, data entry, and custom user interfaces.

Fig 3.8 4x4 KEYPAD

3.9 BLUETOOTH MODULE

The Bluetooth mechanism is one of the earliest wireless technologies that has revolutionized the communication process on devices. This technology works only in the equipment that was designed to do so during the manufacturing and design process, and the modules fit into the internal structure. In modern digital communication, Bluetooth modules are vital components and are incorporated into several devices to provide the easy transfer of data from one device to another. "A Bluetooth module is a specialized chip that is designed to wirelessly connect two compatible devices for communication, and it does it using its low energy wave feature."
These modules act as the interface between the microcontroller and the devices. The scope of these modules is not just limited to smartphones and laptops but also includes multiple types of devices, such as watches and households that use Bluetooth modules to share data.
Bluetooth beacons are used to represent the start, end and mid-point which will be used by the robot to differentiate the specified locations. Bluetooth beacons will act as a terminal for the robot to detect. The autonomous robot will move on an undesignated path i.e. an unmarked route unlike the archaic robots only capable of moving inside a marked black lane. The robot will be trained and multiple simulations are run on it with the help of the DonkeyCar[1] library so it can successfully avoid obstacles and relay the path with efficacy, making the delivery fast and efficient. A Pi camera is used which will help rectify the unmarked desired path over which the robot has to move.

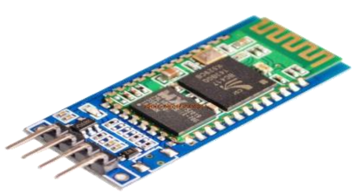


Fig 3.9 BLUETOOTH MODULE

SOFTWARE REQUIREMENTS

3.2 System Setup and Configurations

Embedded programming plays a crucial role in the operation of an autonomous delivery robot by integrating hardware components with software logic to enable navigation, authentication, obstacle avoidance, and secure package handling. The programming is typically done using C++, Python, or embedded C, depending on the microcontroller or single-board computer used.

Setting up and configuring these software components requires a few steps. Here's a general guide:

- Microsoft Windows XP
- Windows7
- Windows10
- Arduino ID

Here’s a breakdown of the setup and configuration for each version of Windows:

a) Microsoft Windows XP Setup & Configuration

1. **Installation**:

Use a bootable CD/DVD with Windows XP installation files.

Boot from the disc and follow the installation wizard.

Select the partition for installation and format it (if needed).

Complete the installation and setup user preferences.

2. **Configuration**:

Install drivers for hardware compatibility.

Apply Service Pack 3 for security updates.

Configure system settings such as network, display, and power options.

Install essential software like antivirus and productivity tools.

b) Microsoft Windows 7 Setup & Configuration

1.Installation:

Use a bootable USB or DVD containing Windows 7 installation files.

Boot from the installation media and follow on-screen instructions.

Choose the partition for Windows 7 installation.

Complete the installation process.

2.Configuration:

Install all necessary drivers and Windows updates.

Configure system settings (display, network, user accounts).

Enable Windows Defender or install third-party security software.

Optimize performance by adjusting startup programs and system settings.

c) Microsoft Windows 10 Setup & Configuration

1.Installation:

Download the Windows 10 ISO or use a bootable USB.

Boot from the installation media and follow setup instructions.

Choose the partition for installation and proceed.

Complete the installation and log in with a Microsoft account (optional).

2.Configuration:

Install necessary drivers and update Windows to the latest version.

Configure privacy settings and enable Windows security features.

Adjust system settings such as display resolution, power options, and notifications.

Install essential software and enable backup options.

d) Arduino IDE Setup & Configuration

1.Download & Install:

Visit the official Arduino website and download the latest IDE version.

Install the software on your system.

2.Connect Your Arduino Board:

Use a USB cable to connect your Arduino board to your computer.

Open the Arduino IDE and select the correct board under Tools > Board.

3.Select the Port:

Navigate to Tools > Port and choose the correct port for your board.

4.Test Your Setup:

Open File > Examples > Basics > Blink and upload the code to verify the setup.

CHAPTER 4

FUNCTIONAL DESCRIPTION

4.1 FLOW CHART

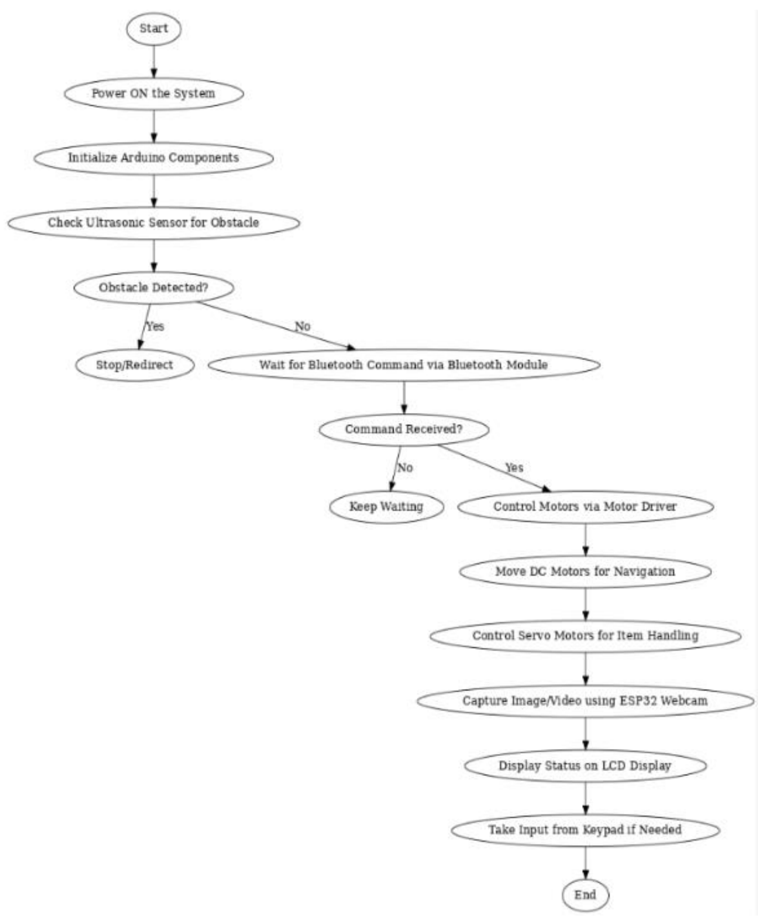


Fig 5.1 FLOW CHART

5.2 WORKING

The process begins with the sender placing a delivery request through a web or mobile application, where the recipient is assigned a unique password or PIN code for authentication. The robot then determines the optimal path using GPS, LiDAR, or camera-based mapping, employing local obstacle avoidance and global path planning to navigate safely. Sensors continuously scan the surroundings to detect obstacles and adjust movement accordingly.

Upon reaching the destination, the delivery compartment remains electronically locked and can only be accessed using the password authentication system. The recipient enters the correct password via a touchscreen, keypad, or mobile app to unlock the compartment. If an incorrect password is entered multiple times, the system triggers a security alert. Throughout the journey, the robot provides live updates to the sender and recipient regarding its location. Unauthorized access attempts activate security measures, including alerts and potential remote shutdown. Additionally, the robot can be equipped with a camera system to record delivery interactions for security purposes.

After successful delivery, the robot either returns to its charging station or proceeds to the next delivery task. If multiple deliveries are scheduled, the robot optimizes its route to minimize travel time. The system is powered by a microcontroller such as an ESP32, Raspberry Pi, or Arduino, which manages various sensors and actuators, including motor drivers, GPS modules, LiDAR, ultrasonic sensors, LCD displays, and servo motors for locking mechanisms. Embedded programming ensures smooth communication between components, enabling autonomous movement, secure authentication, and real-time monitoring. This technology enhances security, efficiency, and automation in package delivery, making it ideal for campuses, offices, and smart cities.

- ➢ User Authentication & Order Placement
  - The sender places a delivery request through a web or mobile application.
  - The recipient is assigned a unique password or PIN code for authentication.
- ➢ Autonomous Navigation

- The robot determines the optimal path using GPS, LiDAR, or camera-based mapping.
- It employs local obstacle avoidance and global path planning to reach the destination safely.

➢ Secure Package Handling

- The delivery compartment is locked electronically and can only be accessed using the password authentication system.
- Upon arrival, the recipient enters the correct password via a touchscreen, keypad, or mobile app to unlock the compartment.

➢ Real-Time Monitoring & Alerts

- The robot sends live updates to the sender and recipient regarding its location.
- If unauthorized access is attempted, an alert is triggered, and security measures are activated.

Return or Next Delivery

successful delivery, the robot returns to its charging station or proceeds to the next delivery task.

5.3 SCHEMATIC DIAGRAM

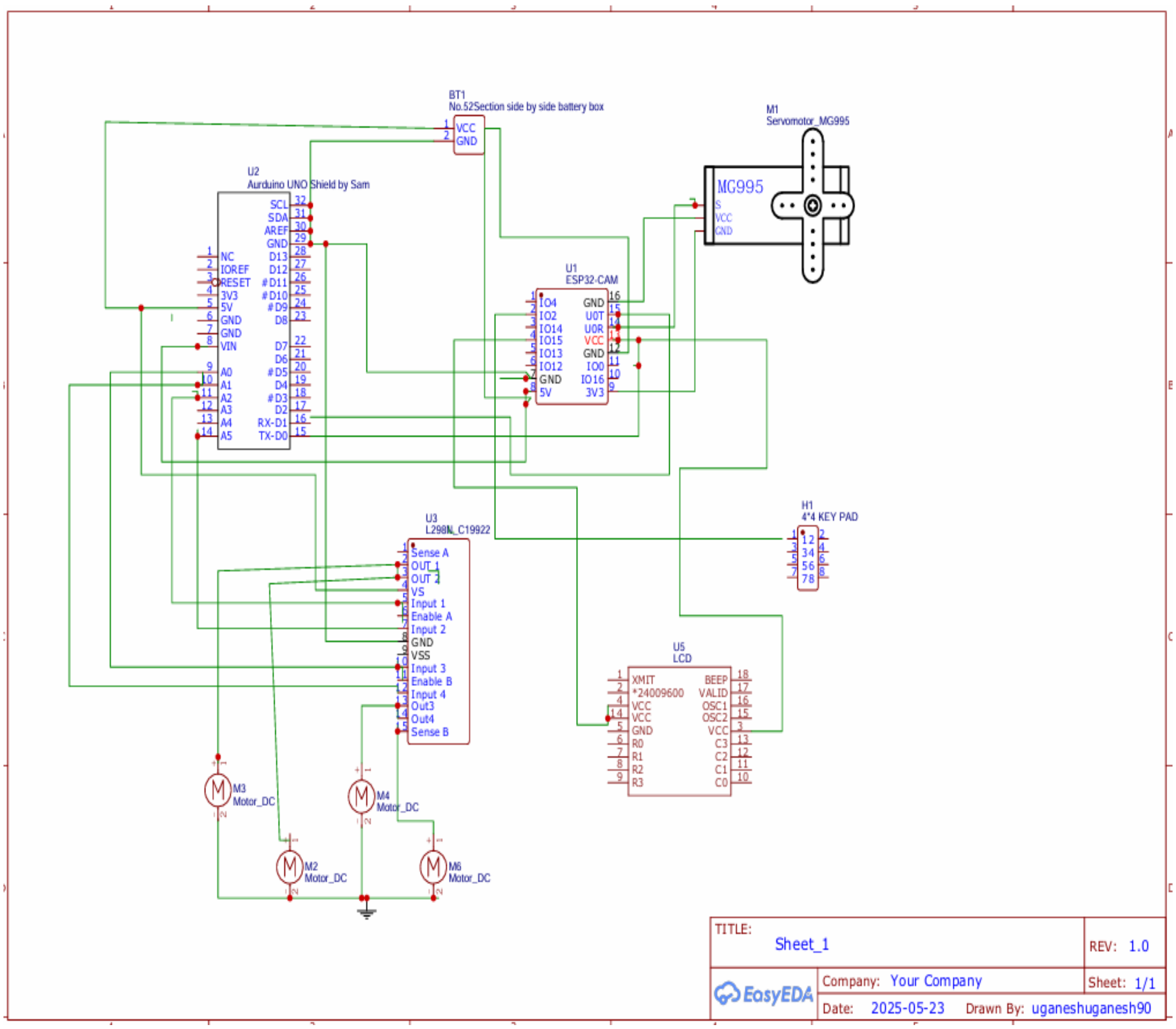


Fig 5.3 SCHEMATIC DIAGRAM

CHAPTER 6

RESULTS

Fig 6.1 RESULT

This system tackles common security threats like impersonation, replay, and man-in-the-middle attacks by combining password authentication with other factors such as audio-visual verification. It represents a significant advancement in secure and efficient package transportation. By integrating AI-driven navigation, real-time tracking, and password-protected access, this system enhances security while ensuring seamless deliveries.

CHAPTER 7

ADVANTAGES

An autonomous delivery robot with password authentication offers several advantages, making it a valuable solution for secure and efficient package delivery. One of its primary benefits is enhanced security

- Real-Time Tracking & Monitoring
- Environmental Benefits
- Versatile Applications
- Improved Accessibility
- Reduced Delivery Errors
- Enhanced Security
- Contactless & Safe Delivery
- Integration with Smart Cities & IoT

DISADVANTAGES

- High Initial Cost
- Dependency on network connectivity
- Battery Limitations

CHAPTER 8

APPLICATIONS

An autonomous delivery robot with password authentication has a wide range of applications across various industries, enhancing security, efficiency, and automation in package delivery. These robots are designed to navigate autonomously, authenticate users securely, and ensure safe transportation of goods.

- ✓ E-Commerce & Logistics
- ✓ Healthcare & Medical Supply Transport
- ✓ Contactless & Fast Delivery
- ✓ Corporate Offices & Document Delivery
- ✓ Smart Cities & Urban Infrastructure
- ✓ Food & Grocery Delivery
- ✓ Educational Institutions
- ✓ Security & Restricted Access Areas
- ✓ Retail & Warehousing
- ✓ Military & Defense Applications
- ✓ Hotel & Hospitality Industry
- ✓ Airport & Luggage Handling
- ✓ Emergency & Disaster Relief

CHAPTER 9

CONCLUSION

The project **“AUTONOMOUS DELIVERY ROBOTS WITH PASSWORD AUTHENTICATION”** has been successfully designed and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. And the project has been successfully implemented.

So at the end it conclude that it represents a significant advancement in secure and efficient package transportation. By integrating AI-driven navigation, real-time tracking, and password-protected access, this system enhances security while ensuring seamless deliveries. The robot’s ability to operate autonomously reduces human dependency, minimizes delivery errors, and optimizes logistics, making it a valuable solution for e-commerce, healthcare, corporate offices, and smart cities.

CHAPTER 10

FUTURE SCOPE

The future scope of an Autonomous Delivery Robot with Password Authentication is vast, with advancements in AI, robotics, and security technologies driving its evolution. Here are some key areas where this project can expand as like AI-Driven Navigation & Smart Path Optimisation which will integrate advanced AI algorithms for real-time path optimization, allowing them to navigate complex environments more efficient and as like IoT & Cloud-Based Monitoring in which the Robots will be connected to cloud-based platforms, enabling remote monitoring, predictive maintenance, and real-time analytics. IoT integration will allow businesses to track deliveries, optimize routes, and enhance security. So the design and implementation of a versatile delivery robot which integrates a secure locking system, GPS navigation, customer authentication, and a camera system for real-time monitoring.

APPENDIX

SOURCE CODING

```
#include "esp_camera.h"
#include <WiFi.h>
#include <ESPmDNS.h>
#include <WebServer.h>

// Replace with your network credentials
const char* ssid = "vivo Y28 5G";
const char* password = "1234uday";

// Optional: mDNS name (access via http://esp32cam.local)
const char* hostname = "esp32cam";

WebServer server(80);

// CAMERA_MODEL_AI_THINKER pin configuration
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(false);
```

```
Serial.println();

// Connect to Wi-Fi
WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

Serial.print("Camera Stream Ready! Go to: http://");
Serial.println(WiFi.localIP());

// Setup mDNS
if (MDNS.begin(hostname)) {
  Serial.print("mDNS responder started at: http://");
  Serial.print(hostname);
  Serial.println(".local");
} else {
  Serial.println("Error setting up mDNS responder!");
}

// Configure camera
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer   = LEDC_TIMER_0;
config.pin_d0       = Y2_GPIO_NUM;
config.pin_d1       = Y3_GPIO_NUM;
config.pin_d2       = Y4_GPIO_NUM;
config.pin_d3       = Y5_GPIO_NUM;
config.pin_d4       = Y6_GPIO_NUM;
config.pin_d5       = Y7_GPIO_NUM;
config.pin_d6       = Y8_GPIO_NUM;
config.pin_d7       = Y9_GPIO_NUM;
config.pin_xclk     = XCLK_GPIO_NUM;
config.pin_pclk     = PCLK_GPIO_NUM;
config.pin_vsync    = VSYNC_GPIO_NUM;
config.pin_href     = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn     = PWDN_GPIO_NUM;
```

```
  config.pin_reset    = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  config.frame_size = FRAMESIZE_QVGA; // change to FRAMESIZE_VGA, etc.
  config.jpeg_quality = 12;
  config.fb_count = 1;

  // Initialize camera
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  // Start web server
  startCameraServer();
}

void loop() {
  server.handleClient();
}

void handle_jpg_stream() {
  WiFiClient client = server.client();
  String response = "HTTP/1.1 200 OK\r\n";
  response += "Content-Type: multipart/x-mixed-replace; boundary=frame\r\n\r\n";
  server.sendContent(response);

  while (1) {
    camera_fb_t *fb = esp_camera_fb_get();
    if (!fb) {
      Serial.println("Camera capture failed");
      return;
    }

    response = "--frame\r\n";
    response += "Content-Type: image/jpeg\r\n\r\n";
    server.sendContent(response);
    server.sendContent((const char*)fb->buf, fb->len);
    server.sendContent("\r\n");
    esp_camera_fb_return(fb);
```

```
    if (!client.connected()) break;
  }
}

void startCameraServer() {
  server.on("/", HTTP_GET, []() {
    server.send(200, "text/html",
    "<html><body><h2>ESP32-CAM Live Stream</h2><img src='/stream'></body></html>");
  });

  server.on("/stream", HTTP_GET, handle_jpg_stream);

  server.begin();
}
```

REFERENCES

[1] Alverhed, E., Hellgren, S., Isaksson, H., Olsson, L., Palmqvist, H., & Flodén, J. (2024).

[2] Shaklab, E., Karapetyan, A., Sharma, A., Mebrahtu, M., Basri, M., Nagy, M., Khonji, M., & Dias, J. (2023).

[3] Yuen, K. F., Cai, L., Lim, Y. G., & Wang, X. (2022). Consumer Acceptance of Autonomous Delivery Robots for Last-Mile Delivery.

[4] Tirkey, S., & Ayyappa, T. (2021). Design and Fabrication of Self-Driving Delivery Robot.

[5] Jennings, D., & Figliozzi, M. (2020). Study of Road Autonomous Delivery Robots and Their Potential Effects on Freight Efficiency and Travel .