A

MAJOR PROJECT REPORT ON

SMART BOX FOR PARCEL COLLECTION

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

AKULA PRASHANTH	**218R1A0401**
AVADHANAM JYOTHIKA NEETU	**218R1A0402**
BABUGONDA PANDARI	**218R1A0403**
BACHANABOINA VAMSHI	**218R1A0404**

Under the Esteemed Guidance of

Mrs. D. LATHA

Assistant Professor, ECE Department

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA & NAAC)
Kandlakoya (V), Medchal (M), Telangana – 501401

2024-2025

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA & NAAC)
Kandlakoya (V), Medchal (M), Telangana – 501401

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Major Project work entitled **"SMART BOX FOR PARCEL COLLECTION"** is being submitted by **A. PRASHANTH** bearing Roll No: **218R1A0401, A. JYOTHIKA NEETU** bearing Roll No**: 218R1A0402, B. PANDARI** bearingRoll No: **218R1A0403, B. VAMSHI** bearing Roll No: **218R1A0404** in B. Tech IV-II semester, Electronics and Communication Engineering is a record bonafide work carried out by then during the academic year 2024-2025.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
Mrs. D. LATHA	Dr. SUMAN MISHRA
Assistant Professor, ECE	Professor, ECE
CMREC	CMREC

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We sincerely thank the management of our college **CMR Engineering College** for providing required facilities during our project work. We derive great pleasure in expressing our sincere gratitude to our principal **Dr. A. S. REDDY** for his timely suggestions, which helped us to complete the project work successfully. It is the very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA,** Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Dr. T. SATYANARAYANA**, Associate Professor, Department of ECE for the ideas that led to complete the project work and we also thank him for his continuous guidance, support and unfailing patience, throughoutthe course of this work. We sincerely thank our project internal guided **Mrs**. **D. LATHA,** Assistant professor of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We hereby declare that the project work entitled **"SMART BOX FOR PARCEL COLLECTION"** is the work done by us in campus at **CMR ENGINEERING COLLEGE,** Kandlakoya during the academic year 2024-2025 and is submitted as Major project in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING FROM JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD.**

AKULA PRASHANTH	**218R1A0401**
AVADHANAM JYOTHIKA NEETU	**218R1A0402**
BABUGONDA PANDARI	**218R1A0403**
BACHANABOINA VAMSHI	**218R1A0404**

ABSTRACT

The increasing volume of online shopping and package deliveries has led to a growing need for secure and convenient parcel collection systems. This project proposes the design and implementation of a Smart Box for Parcel Collection, which utilizes RFID (Radio Frequency Identification) technology and QR code scanning to enable secure and automated access to delivered parcels. The system integrates an RFID reader for automatic identification of parcels tagged with RFID tags and a QR code scanning mechanism for user authentication and access.

The proposed smart box functions by allowing delivery personnel to place parcels inside secure compartments. Each parcel is identified through its unique RFID tag, and recipients are notified of the delivery with a QR code or access code for authentication. Users can retrieve their parcels by scanning the QR code or using an RFID-enabled card, which triggers the unlocking of the corresponding compartment.

By leveraging the capabilities of RFID and QR code technologies, the smart box ensures secure, efficient, and contactless parcel retrieval, eliminating the need for manual handling and reducing the risk of package theft or misplacement. The system is scalable and can be deployed in various environments such as smart homes, apartment complexes, offices, and logistics hubs, providing a versatile solution to modern parcel delivery challenges.

This project explores the hardware and software components required to implement the system, including RFID tags, readers, electronic locks, mobile apps, and cloud-based management tools. The solution aims to enhance user convenience, improve delivery efficiency, and provide a seamless, automated experience for both recipients and delivery services.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The A Smart Box for Parcel Collection that utilizes RFID (Radio Frequency Identification) and QR Code technologies is an advanced solution designed to streamline and secure the process of receiving and retrieving parcels. In this system, parcels are tagged with unique RFID tags that can be scanned by an RFID reader when they are dropped off in the smart box. The RFID technology allows for real-time tracking of the parcel as it moves through the system, ensuring accurate and contactless identification without the need for manual scanning. Once the parcel is placed in the box, the system generates a unique QR Code linked to the specific parcel. This QR code can be used by the recipient to retrieve their parcel by scanning it at the Smart Box. The QR code acts as an authentication tool, ensuring that only authorized individuals can access the correct parcel.

This combination of RFID for tracking and QR codes for secure pickup makes the system efficient, user-friendly, and scalable. Users can pick up their parcels at their convenience, and the system ensures high levels of security and privacy. Additionally, the system may offer notifications and data analytics for parcel management and tracking, making it a versatile solution for e-commerce, office spaces, and community delivery points. Overall, this hybrid RFID and QR code solution provides a seamless, secure, and automated way to manage parcel deliveries and collections.

1.2 OBJECTIVE OF THE PROJECT

The primary objective of a Smart Box for Parcel Collection using RFID and QR Code technology is to revolutionize the way parcels are delivered, tracked, and retrieved. By automating the process, this system aims to enhance both efficiency and security in parcel management. The use of RFID technology enables real-time tracking of parcels, ensuring that items are accurately identified and monitored throughout their journey, from drop-off to pick up. This contactless technology eliminates the need for manual scanning or handling, reducing human error and speeding up the parcel processing time. On the other hand, the integration of QR codes allows recipients to easily retrieve their parcels by scanning a unique code linked to their package. This improves the user experience by offering 24/7 access to parcels, without the need for staff interaction or physical verification, making it ideal for busy urban environments or communities. The system also enhances security by ensuring that only authorized recipients can open the designated box containing their parcel, providing an added layer of protection against theft or unauthorized access.

Another core objective is to create a scalable and cost-effective solution for parcel collection and delivery. As e-commerce continues to grow, there is an increasing demand for efficient and accessible parcel management systems. The Smart Box addresses this need by being adaptable to various environments, whether it's a residential apartment complex, office building, or a public hub. With multiple compartment sizes and the ability to integrate with different delivery service providers, the system is designed to handle a wide variety of parcel sizes and quantities, making it a versatile solution for businesses and consumers alike. Additionally, the system reduces operational costs associated with traditional delivery models, such as the need for delivery personnel or physical customer service desks. By automating parcel collection and retrieval, it frees up resources and reduces the risk of errors or delays, ensuring a seamless experience for both senders and recipients. Ultimately, the objective of the Smart Box is to provide a secure, convenient, and scalable solution that enhances the logistics and parcel delivery process, offering a modern alternative to outdated and inefficient systems.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

1.Existing Parcel Lockers (Manual):

In some locations, especially urban areas, parcel lockers have become a popular alternative. These lockers, which can be found in shopping malls, transport hubs, or apartment complexes, allow delivery personnel to deposit parcels securely in a designated locker. The recipient is then notified through a code or access link to open the locker and collect their parcel. While this system offers increased security compared to doorstep delivery, it has its drawbacks. Availability can be limited, as lockers may be fully occupied, particularly during peak shopping periods. Additionally, manual authentication methods (such as entering a code on a keypad or scanning a barcode) can be prone to human error or delays, which reduces the overall user experience. Moreover, real-time parcel tracking is often not integrated into these systems, so recipients may have to wait for a physical notification to know when their parcel is available for pickup.

2. Post Offices or Designated Pickup Points:

Another common method is for parcels to be delivered to a local post office or designated pickup point, where recipients can collect their packages. Once the parcel arrives at the post office, the recipient receives a notification and must visit the location during business hours to retrieve it. This method is widely used by courier companies and postal services but comes with its own set of challenges. Inconvenience is a significant drawback, as recipients must travel to the pickup point, which may not always be close to their home or office. Moreover, many pickup points have limited hours of operation, making it difficult for individuals with busy schedules to retrieve their parcels. In high-demand locations, such as during the holiday season, recipients may face long wait times or delays in retrieving their packages. This method also lacks flexibility and can lead to inefficiencies in the parcel collection process, particularly for those who need their packages urgently.

PROPOSED SYSTEM

The proposed method introduces an automated smart box system that leverages RFID technology and QR codes to streamline and secure the parcel collection process. The system provides a more efficient, secure, and flexible solution for parcel deliveries.

1. RFID-Based Parcel Identification:
    - RFID tags are attached to parcels during shipping, each containing a unique identifier linked to the parcel's delivery details. The recipient's smart box includes an RFID reader that scans the parcel's RFID tag upon placement.
    - Automation: As soon as the parcel is scanned, the system logs the delivery, assigns it to a specific compartment, and updates the recipient via a mobile app or SMS notification.
    - Security: RFID technology ensures that only the correct recipient can retrieve their parcel, reducing the chance of mis delivery or theft.
2. QR Code for User Authentication:
    - Upon receiving a notification, the recipient uses a QR code generated through the app or email to authenticate their identity. This code can either be scanned from their mobile device or used in conjunction with their RFID card.
    - Access Control: When the recipient scans the QR code at the smart box, the system authenticates the user and unlocks the corresponding locker compartment, allowing them to retrieve their parcel.
    - User Experience: The process is entirely contactless, reducing manual handling and offering a streamlined, secure experience for the recipient.
3. Smart Box Components:
    - The smart box is equipped with electronic locks, a touchscreen display, and wireless connectivity to manage parcel storage, retrieval, and notifications.
    - Real-Time Updates: The smart box can send instant updates to users, notifying them when their parcel has been placed inside the box and when it is ready for pickup.
    - 24/7 Access: Recipients can access their parcels at any time of day, even outside business hours, providing greater flexibility than traditional delivery methods.

Improved Parcel Management:

The system is designed to efficiently manage multiple parcels for different users within a single smart box unit. Dynamic allocation of lockers based on parcel size and recipient information ensures an optimized use of space.

- Scalability: The solution can be expanded to larger buildings or complexes, enabling secure and automated parcel management for multiple users simultaneously.

Comparison of Existing and Proposed Methods:

Feature	Existing Methods	Proposed Method
Delivery Process	Manual delivery to doorsteps, pickup points, or post offices.	Automated delivery to secure, RFID-enabled lockers.
Security	Limited security; parcels can be left unattended or at risk of theft.	High security through RFID tags and QR code authentication.
User Interaction	Requires physical interaction, manual tracking, or keypads.	Contactless interaction with RFID/QR code for authentication.
Convenience	Limited hours for pickup, potential for missed deliveries.	24/7 access for parcel retrieval, reducing the need for rescheduling.
Tracking	Parcel tracking is manual or external (e.g., via third-party apps).	Real-time tracking updates and notifications directly to users.
Scalability	Limited to available locker space or pickup points.	Scalable to large buildings, apartment complexes, or businesses.
Automation	Requires manual handling, redelivery, or third-party involvement.	Fully automated with minimal human intervention.
Cost	Varies by method (may require multiple delivery attempts, manual handling).	Initial setup cost, but more efficient long-term with reduced need for manual intervention.

Table 2.1: Comparison of Existing and Proposed Method

2.2 EMBEDDED SYSTEMS INTRODUCTION

An embedded system is a combination of computer hardware and software designed for a specific function or functions within a larger system. The systems can be programmable or with fixed functionality. Industrial machines, consumer electronics, agricultural and process industry devices, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys, as well as mobile devices, are possible locations for an embedded system. While embedded systems are computing systems, they can range from having no user interface (UI) -- for example, on devices in which the system is designed to perform a single task -- to complex graphical user interfaces (GUIs), such as in mobile devices.

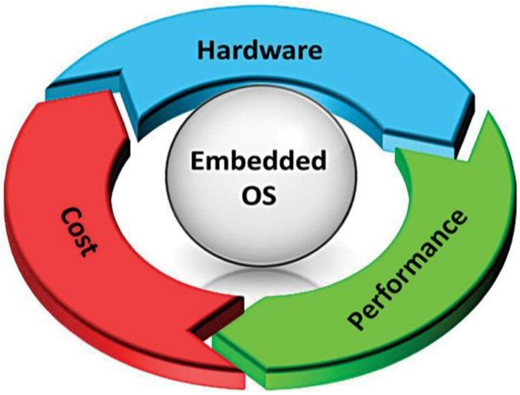

Fig:2.1 Embedded OS

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit (IC) in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect real-time flight data.

In 1965, Autonoetic, now a part of Boeing, developed the D-17B, the computer used in the Minuteman I missile guidance system. It is widely recognized as the first mass-produced embedded system. When the Minuteman II went into production in 1966, the D-17B was replaced with the NS-17 missile guidance system, known for its high-volume use of integrated circuits. In 1968, the first embedded system for a vehicle was released; the Volkswagen 1600 used a microprocessor to control its electronic fuel injection system.

By the late 1960s and early 1970s, the price of integrated circuits dropped, and usage surged. The first microcontroller was developed by Texas Instruments in 1971. The TMS 1000 series, which became commercially available in 1974, contained a 4-bit processor, read-only memory (ROM) and random-access memory (RAM), and cost around $2 apiece in bulk orders.

Also, in 1971, Intel released what is widely recognized as the first commercially available processor, the 4004. The 4-bit microprocessor was designed for use in calculators and small electronics, though it required eternal memory and support chips. The 8-bit Intel 8008, released in 1972, had 16 KB of memory; the Intel 8080 followed in 1974 with 64 KB of memory. The 8080's successor, x86 series, was released in 1978.

In 1987, the first embedded operating system, the real-time VxWorks, was released by Wind River, followed by Microsoft's Windows Embedded CE in 1996. By the late 1990s, the first embedded Linux products began to appear. Today, Linux is used in almost all embedded devices.

Characteristics of embedded systems

The main characteristic of embedded systems is that they are task specific. They perform a single task within a larger system. For example, a mobile phone is not an embedded system, it is a combination of embedded systems that together allow it to perform a variety of general-purpose tasks. The embedded systems within it perform specialized functions. For example, the GUI performs the singular function of allowing the user to interface with the device. In short, they are programmable computers, but designed for specific purposes, not general ones.

The hardware of embedded systems is based around microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers, and generally refer to a CPU that is integrated with other basic computing components such as memory chips and digital signal processors (DSP). Microcontrollers have those components built into one chip.

Embedded systems are specialized, resource-constrained computing systems designed to perform specific functions within larger devices or applications. They are typically characterized by their dedicated purpose, real-time operation, and efficient use of resources, such as limited processing power, memory, and energy consumption.

The embedded systems within it perform specialized functions. For example, the GUI performs the singular function of allowing the user to interface with the device. In short, they are programmable computers, but designed for specific purposes, not general ones.

Embedded systems are specialized, resource-constrained computing systems designed to perform specific functions within larger devices or applications. They are typically characterized by their dedicated purpose, real-time operation, and efficient use of resources, such as limited processing power, memory, and energy consumption.

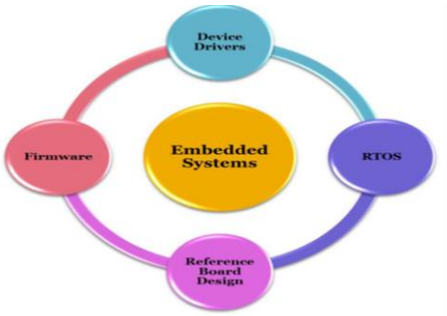

Fig:2.2 Embedded Systems

Additionally, embedded systems can include the following characteristics:

- comprised of hardware, software and firmware.
- embedded in a larger system to perform a specific function as they are built for specialized tasks within the system, not various tasks.
- either microprocessor-based or microcontroller-based -- both are integrated circuits that give the system compute power.
- often used for sensing and real-time computing in internet of things (IoT) devices -- devices that are internet-connected and do not require a user to operate.
- vary in complexity and in function, which affects the type of software, firmware and hardware they use; and xix
- often required to perform their function under a time constraint to keep the larger system functioning properly.

Embedded systems vary in complexity, but generally consist of three main elements:

- **Hardware** of embedded systems is based around microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers, and generally refer to a CPU that is integrated with other basic computing components such as memory chips and digital signal processors (DSP). Microcontrollers have those components built into one chip.
- **Software** for embedded systems can vary in complexity. However, industrial-grade

microcontrollers and embedded IoT systems generally run very simple software that requires little memory.

- **Firmware** Embedded firmware is usually used in more complex embedded systems to connect the software to the hardware. Firmware is software that interfaces directly with the hardware. A simpler system may just have software directly in the chip, but more complicated systems need firmware under more complex software applications and operating systems.

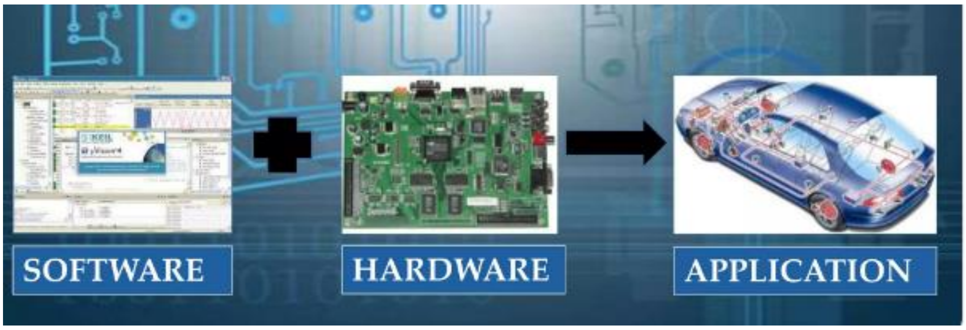

Fig:2.3 Blocks of Embedded Systems

2.3WHY EMBEDDED SYSTEMS?

An embedded system is a computer system with a particular defined function within a larger mechanical or electrical system. They control many devices in common use. They consume low power, are of a small size and their cost is low per unit.

Modern embedded systems are often based on micro-controllers. A micro-controller is a small computer on a single integrated circuit which contains a processor core, memory, and programmable input and output peripherals. As Embedded system is dedicated to perform specific tasks therefore, they can be optimized to reduce the size and cost of the product and increase the reliability and performance. Almost every Electronic Gadget around us is an Embedded System, digital watches, MP3 players, Washing Machine, Security System, scanner, printer, a cellular phone, Elevators, ATM, Vendor Machines, GPS, traffic lights, Remote Control, Microwave Oven and many more. The uses of embedded systems are virtually limitless because every day new products are introduced to the market which utilize embedded computers in several ways.

Let's make it easy for you. For Example – You are sitting in a train headed to your destination and you are already fifty miles away from your home and suddenly you realize that you forgot to switch of the fan. Not to worry, you can switch it off just by clicking a button on your cell phone using this technology – The Internet of Things. Well, this is just one good thing about IoT. We can monitor Pollution Levels, we can control the intensity of streetlights as per the season and weather requirements, IoT can also provide the parents with real-time information about their baby's breathing, skin temperature, body position, and activity level on their smartphones and many other applications which can make our life easy.

Embedded Systems has brought about a revolution in science. It is also a part of an Internet of Things (IoT) – a technology in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Let's make it easy for you. For Example – You are sitting in a train headed to your destination and you are already fifty miles away from your home and suddenly you realize that you forgot to switch of the fan. Not to worry, you can switch it off just by clicking a button on your cell phone using this technology – The Internet of Things. Well, this is just one good thing about IoT. We can monitor Pollution Levels, we can control the intensity of streetlights as per the season and weather requirements, IoT can also provide the parents with real-time information about their baby's breathing, skin temperature, body position, and activity level on their smartphones and many other applications which can make our life easy.

Fig:2.4 Embedded System Hardware

2.4 DESIGN APPROACHES

A system designed with the embedding of hardware and software together for a specific function with a larger area is embedded system design. In embedded system design, a microcontroller plays a vital role. Micro-controller is based on Harvard architecture, it is an important component of an embedded system. External processor, internal memory and i/o components are interfaced with the microcontroller. It occupies less area, less power consumption. The applications of microcontrollers is MP3, washing machines.

Critical Embedded Systems (CES) are systems in which failures are potentially catastrophic and, therefore, hard constraints are imposed on them. In the last years the amount of software accommodated within CES has considerably changed. This change means that software design for these systems is also bounded to hard constraints (e.g., high security and performance). Along with the evolution of CES, the approaches for designing them are also changing rapidly, to fit the specialized needs of CES. Thus, a broad understanding of such approaches is missing.

Steps in the Embedded System Design Process

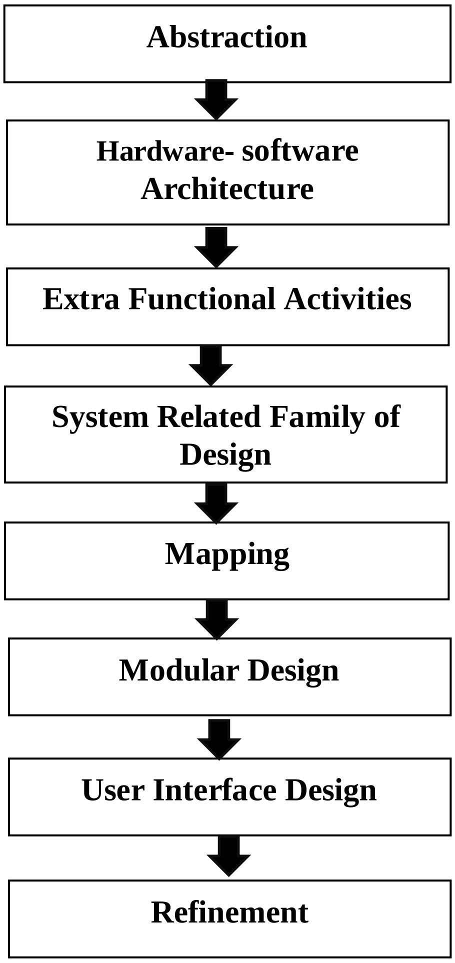

Fig:2.5 Embedded Design Process Steps

Abstraction

In this stage the problem related to the system is abstracted.

Hardware – Software Architecture

Proper knowledge of hardware and software to be known before starting any design process.

Extra Functional Properties

Extra functions to be implemented are to be understood completely from the main design.

System Related Family of Design

When designing a system, one should refer to a previous system-related family of design.

Modular Design

Separate module designs must be made so that they can be used later on when required.

Mapping

Based on software mapping is done. For example, data flow and program flow are mapped into one.

User Interface Design

In user interface design it depends on user requirements, environment analysis and function of the system. For example, on a mobile phone if we want to reduce the power consumption of mobile phones, we take care of other parameters, so that power consumption can be reduced.

Refinement

Every component and module must be refined appropriately so that the software team can understand.

Architectural description language is used to describe the software design.

- Control Hierarchy
- Partition of structure
- Data structure and hierarchy
- Software Procedure.

In user interface design it depends on user requirements, environment analysis and function of the system. For example, on a mobile phone if we want to reduce the power consumption of mobile phones, we take care of other parameters, so that power consumption can be reduced. To help countries and health-care facilities to achieve system change and adopt alcohol-based hand rubs as the gold hand hygiene in health care, WHO has identified formulations for their local preparation. Logistic, economic, safety, and cultural.

TABLE:2.2 Embedded System Design Software Development Activities

Design Metrics / Design Parameters of an Embedded System	Function
Power Dissipation	Always maintained low
Performance	Should be high
Process Deadlines	The process/task should be completed within a specified time.
Manufacturing Cost	Should be maintained.
Engineering Cost	It is the cost for the edit-test-debug of hardware and software.
Size	Size is defined in terms of memory RAM/ROM/Flash Memory/Physical Memory.
Prototype	It is the total time taken for developing a system and testing it.
Safety	System safety should be taken like phone locking, user safety like engine breaks down safety measure must be taken
Maintenance	Proper maintenance of the system must be taken, in order to avoid system failure.
Time to market	It is the time taken for the product/system developed to be launched into the market.

Embedded systems are used in a variety of technologies across industries. Some examples include:

- **Automobiles** Modern cars commonly consist of many computers (sometimes as many as 100), or embedded systems, designed to perform different tasks within the vehicle. Some of these systems perform basic utility function and others provide entertainment or user-facing functions. Some embedded systems in consumer vehicles include cruise control, backup sensors, suspension control, navigation systems and airbag systems.

- In user interface design it depends on user requirements, environment analysis and function of the system. For example, on a mobile phone if we want to reduce the power consumption of mobile phones, we take care of other parameters, so that power consumption can be reduced. To help countries and health-care facilities to achieve system change and adopt alcohol-based hand rubs as the gold hand hygiene in health care, WHO has identified formulations for their local preparation. Logistic, economic, safety, and cultural.

- Some of these systems perform basic utility function and others provide entertainment or user-facing functions. Some embedded systems in consumer vehicles include cruise control, backup sensors, suspension control, navigation systems and airbag systems.

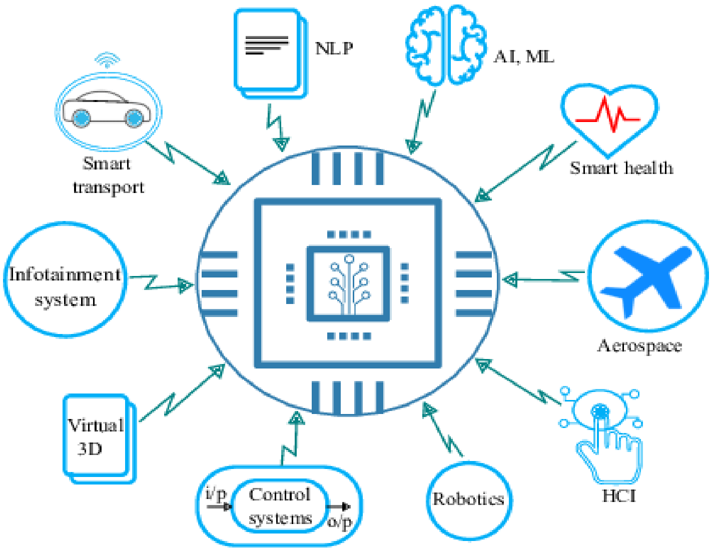

Fig:2.6 Applications of Embedded Systems

- **Mobile phones** consist of many embedded systems, including GUI software and hardware, operating systems, cameras, microphones and USB I/O modules.
- **Industrial machines** can contain embedded systems, like sensors, and can be embedded systems themselves. Industrial machines often have embedded automation systems that perform specific monitoring and control functions.

- **Medical equipment** These may contain embedded systems like sensors and control mechanisms. Medical equipment, such as industrial machines, also must be very user-friendly, so that human health isn't jeopardized by preventable machine mistakes. This means they'll often include a more complex OS and GUI designed for an appropriate UI.

The choice of components for the WHO-recommended hand rub formulations takes into account cost constraints and microbicidal activity. The following two formulations are recommended for local production with a maximum of 50 litres per lot to ensure safety in production and storage.

2.5 COMBINATION OF LOGIC DEVICES

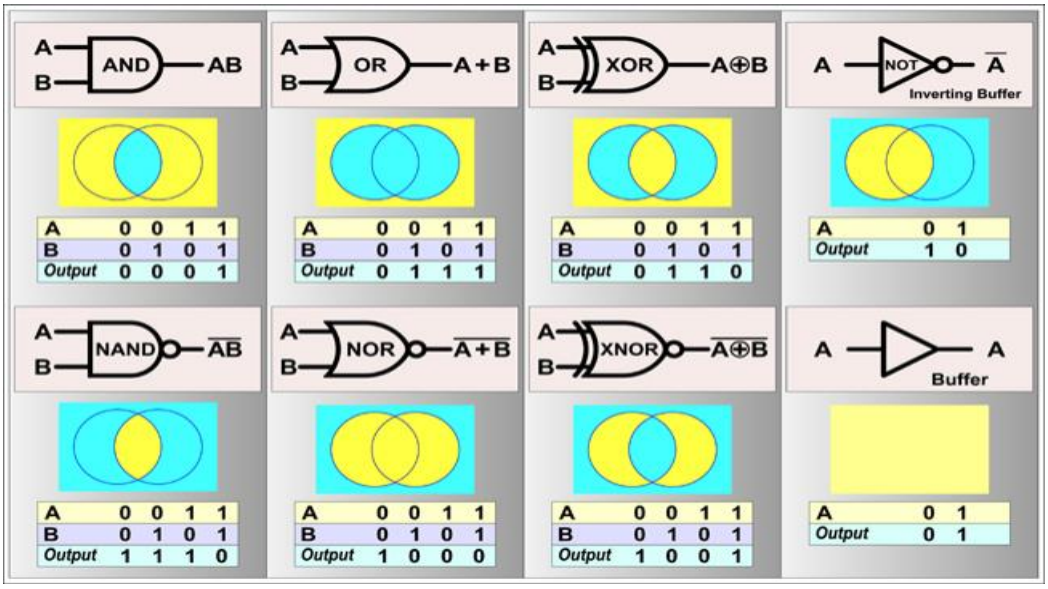

Fig:2.7 Logic Gates

Logic gates are physical devices that use combinational logic to switch an electrical one ("1") or zero ("0") to downstream blocks in digital design. Combinational logic uses those bits to send or receive data within embedded systems. Data bits build into digital words used to communicate with other design blocks within the system. Digital bits and words do this with logic gates in an organized fashion using dedicated address, data, or control signal nodes. Logic gates are the physical devices that enable processing of many 1's and 0's.

Logic families are collections of integrated circuits containing logic gates that perform functions needed by embedded systems to communicate with one another to drive the design. Logic gates are organized into families relative to the type of material and its operational characteristics.

Most logic gates are made from silicon, although some utilize gallium arsenide or other semiconductor materials. The semiconductor material is doped for organization into layers. The doped layers drive power capabilities and typical impedances at input or outputs of each gate. Logic gates used together must employ the same, or complementary, material properties. Knowledge of material properties for logic gates will drive selection of parts within design blocks.

Embedded systems' evolution was built from combinational logic families made possible from the discovery of the transistor. The transistor is made from semiconductor material and is compact. It is able to handle large amounts of power quickly. The transistor employs three terminals to activate electron flow for use in downstream devices as electricity.

Electricity represented as 1's and 0's combines to communicate information throughout an embedded system. Because of its compact size, many millions of transistors combine within very small spaces. This allows millions of gates to operate in compact areas while transmitting and receiving mind-boggling amounts of intelligence through combinational logic. This is all accomplished within a minimal power budget.

Fig:2.8 Embedded System Group

CHAPTER 3
HARDWARE REQUIREMENTS

3.1 HARDWARE

Embedded system hardware

Embedded system hardware can be microprocessor or microcontroller based. In either case, an integrated circuit is at the heart of the product that is generally designed to carry out real time computing. Microprocessors are visually indistinguishable from microcontrollers. However, the microprocessor only implements a central processing unit (CPU) and, thus, xxviii requires the addition of other components such as memory chips. Conversely, microcontrollers are designed as self-contained systems.

Microcontrollers include not only a CPU, but also memory and peripherals such as flash memory, RAM or serial communication ports. Because microcontrollers tend to implement full (if relatively low computer power) systems, they are frequently used on more complex tasks. For example, microcontrollers are used in the operations of vehicles, robots, medical devices and home appliances. At the higher end of microcontroller capability, the term System on a chip (SOC) is often used, although there's no exact delineation in terms of RAM, clock speed, power consumption and so on.

It is one of the characteristics of embedded and cyber-physical systems that both hardware and software must be taken into account. The reuse of available hard- and software components is at the heart of the platform-based design methodology. Consistent with the need to consider available hardware components and with the design information flow, we are now going to describe some of the essentials of embedded system hardware.

Hardware for embedded systems is much less standardized than hardware for personal computers. Due to the huge variety of embedded system hardware, it is impossible to provide a comprehensive overview of all types of hardware components. Nevertheless, we will try to provide a survey of some of the essential components which can be found in most systems.

The choice of components for the WHO-recommended hand rub formulations takes into account cost constraints and microbicidal activity. The following two formulations are recommended for local production with a maximum of 50 litres per lot to ensure safety in production and storage. Nevertheless, we will try to provide a survey of some of the essential components which can be found in most systems.

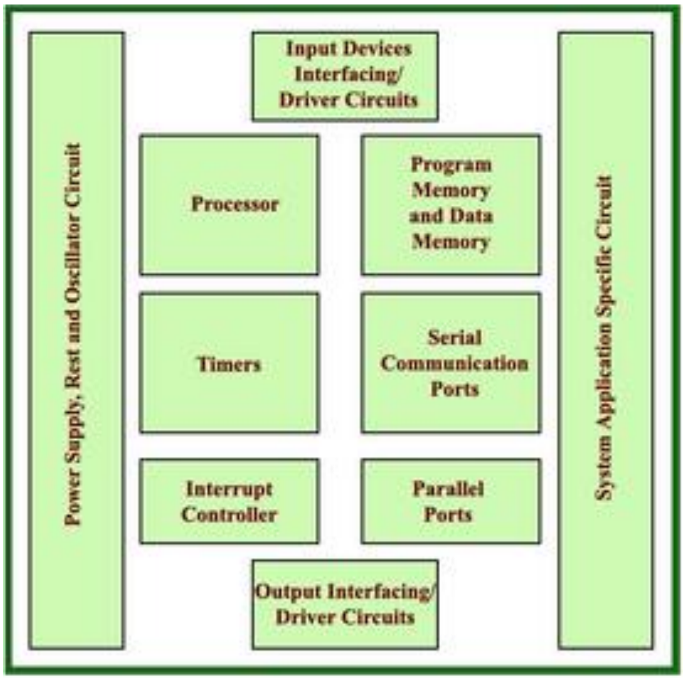

Fig:3.1 Embedded System Hardware

Markets and Markets, a business to business (B2B) research firm, predicts that the embedded market will be worth $116.2 billion by 2025. Chip manufacturers for embedded systems include many well-known technology companies, such as Apple, IBM, Intel and Texas Instruments, as well as numerous other companies less familiar to those outside the field. The expected growth is partially due to the continued investment in artificial intelligence (AI), mobile computing and the need for chips designed for that high-level processing.

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

Often manufacturers of games will provide the consumer with a set of requirements that are different from those that are needed to run a software. These requirements are usually called the recommended requirements. These requirements are almost always of a significantly higher level than the minimum requirements, and represent the ideal situation in which to run the software. Generally speaking, this is a better guideline than minimum system requirements in order to have a fully usable and enjoyable experience with that software.

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. A second meaning of the term of system requirements, is a generalisation of this first definition, giving the requirements to be met in the design of a system or subsystem.

Architecture

All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures.

Processing Power

The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speed often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity, and are often mentioned in this category.

Memory

All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

Secondary Storage

Data storage device requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

Display Adapter

Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters in the system requirements.

Peripherals

Some software applications need to make extensive and/or special use of some peripherals, demanding the higher performance or functionality of such peripherals. Such peripherals include CD-ROM drives, keyboards, pointing devices, network devices, etc

Basic Structure Of An Embedded System

The following illustration shows the basic structure of an embedded system –

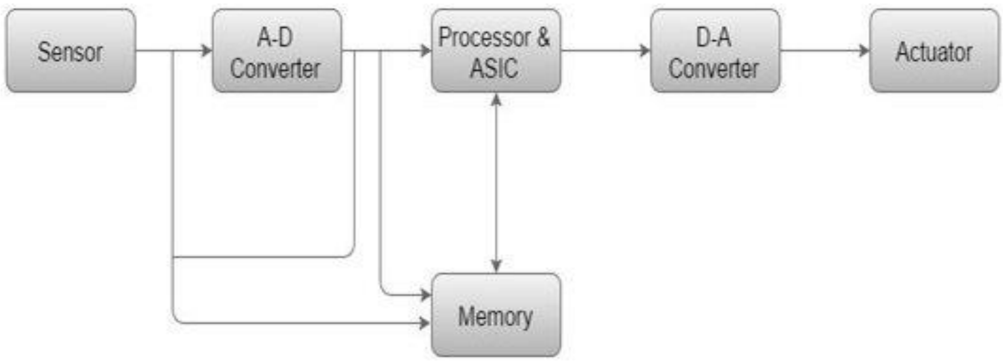

Fig:3.2 Basic Embedded Structure

- **Sensor** − It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** − An analog-to-digital converter converts the analog signal sent by

the sensor into a digital signal.

- Processor & ASICs − Processors process the data to measure the output and store it to the memory.
- **D-A Converter** − A digital-to-analog converter converts the digital data fed by the processor to analog data
- **Actuator** − An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

In this loop, information about the physical environment is made available through sensors. Typically, sensors generate continuous sequences of analog values. In this book, we will restrict ourselves to information processing where digital computers process discrete sequences of values. Appropriate conversions are performed by two kinds of circuits: sample-and-hold-circuits and analog-to-digital (A/D) converters. After such conversion, information can be processed digitally. Generated results can be displayed and also be used to control the physical environment through actuators. Since most actuators are analog actuators, conversion from digital to analog signals is also needed. This model is obviously appropriate for control applications. For other applications, it can be employed as a first order approximation. In the following, we will describe essential hardware components of cyber-physical systems following the loop structure

- Platform limitations. Cross compilers, linkers, auto-code generators, etc.
- Choice of operating environment, such as bare metal minimal kernel, real time operating system, or regular operating system.
- Whether you can rely on your tools. In particular, you need to understand how your tools can fail, and how you'd know if something went wrong.
- Communications protocols. Synchronous and asynchronous communications. Error checking. Error correcting.
- Timing issues, clock rates, reentrancy. Anything that might stop you meeting your real time response targets, or impact the firing of timers.
- Exception handling.
- Interrupts. How to handle them. How long they take. What the impact is upon responding to real time response targets.
- How to test on the platform you are working with.

As well as that, you also need to know how to code. Choose a language well and constrain your use of it to assure you meet the limitations you found by considering the above. If required for the domain, use static code checking. If necessary, use formal proof.

3.2 INTRODUCTION TO RFID TAG

The **RFID tag** is a small, passive electronic device that contains a unique identifier for each parcel. It is attached to the parcel during the drop-off process, where it acts as a marker that identifies the parcel in the system. The tag uses **radio frequency** to communicate with the **RFID reader**, enabling the system to track the parcel's movement and location within the Smart Box. These tags do not require a power source, making them cost-effective and reliable for identifying parcels in real-time. They can store data such as parcel ID, arrival time, and sender details, which are then linked to the system's cloud or database for ongoing tracking and management.

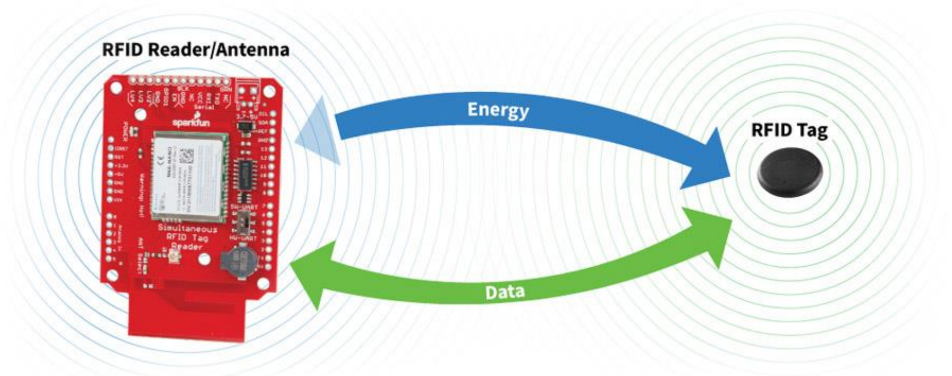

FIG: 3.2 RFID Reader And Tag

3.3RFID READER

The **RFID reader** is a key component that scans the RFID tags attached to the parcels. It emits a radio frequency signal that activates the RFID tag and reads the data stored within it. When a parcel is placed in the Smart Box, the RFID reader detects the unique tag ID and sends this information to the **microcontroller** or the central processing unit. The reader’s main function is to ensure accurate identification and tracking of parcels as they enter or leave the system.

The working principle of an RFID reader revolves around electromagnetic induction or radio wave backscatter. When an RFID tag enters the reader's field, it receives energy from the emitted radio waves. Passive tags, which lack their own power source, rely on this energy to transmit data back to the reader. Active tags, however, contain a built-in battery that allows them to communicate over longer distances. The reader then captures and processes this data, which can include unique identification numbers, product details, or access credentials. This wireless method of communication significantly reduces manual labor and minimizes errors in data entry, improving operational efficiency in industries that require large-scale asset tracking.

Security and privacy concerns are important considerations when implementing RFID systems. Since RFID readers can capture data wirelessly, they are vulnerable to eavesdropping, cloning, and unauthorized access. To mitigate these risks, organizations implement encryption, authentication protocols, and secure access control measures. Some advanced RFID readers support cryptographic techniques to prevent data breaches, ensuring that only authorized personnel can access the information. Additionally, RFID blocking technologies and shielding materials can be used to prevent unauthorized scanning of RFID tags in sensitive applications, such as contactless payments and identity verification systems.

The RFID reader is typically integrated with the Smart Box system and is connected to the database to update parcel status in real-time. Once the parcel is tagged, the sender places it in one of the compartments of the Smart Box. The RFID reader in the Smart Box scans the parcel's RFID tag, capturing the unique ID and registering the parcel in the system. The system then generates a unique QR Code linked to the parcel, containing important tracking and pickup details. This QR code is sent to the recipient via email, SMS, or a mobile app, notifying them that the parcel is ready for collection.

FIG: 3.3 RFID Reader

3.4 QR CODE GENERATOR

The **QR Code generator** is responsible for creating a unique QR Code for each parcel once it is registered in the system. After an RFID tag is scanned, the system generates a QR code linked to the parcel's unique identifier and tracking details. The QR Code typically contains the parcel ID, authentication information, and a tracking link that enables the recipient to retrieve their parcel securely. This QR code is then sent to the recipient via email, SMS, or mobile app. The recipient uses the QR Code to authenticate their access to the Smart Box when they arrive to pick up their parcel. QR Codes are a fast, reliable, and user-friendly method for accessing parcel information and enhancing security during pickup.

A **QR (Quick Response) Code Generator** is a tool used to create QR codes, which are two-dimensional barcodes capable of storing a variety of information, such as URLs, contact details, payment information, and more. Unlike traditional barcodes, which can only hold limited numerical data, QR codes can store alphanumeric characters, special symbols, and even binary data. These codes can be scanned using smartphones, tablets, or dedicated QR code scanners, making them highly versatile for digital transactions, marketing, authentication, and access control. The ease of generating and scanning QR codes has led to their widespread adoption in both commercial and personal applications.

The process of generating a QR code involves encoding data into a pattern of black squares arranged on a white background. A QR Code Generator converts input data—such as text, URLs, or files—into a structured format using a special algorithm called **Reed-Solomon error correction**. This feature ensures that QR codes remain scannable even if they are partially damaged or obscured. QR codes can also be customized with different colors, logos, and design elements to align with branding requirements while maintaining their scannability. Various online and offline QR code generator tools allow users to create static or dynamic QR codes, depending on their needs.

QR codes are generally categorized into **static** and **dynamic** types. **Static QR codes** have fixed information and cannot be changed once generated. They are useful for business cards, product packaging, and event tickets. **Dynamic QR codes**, on the other hand, allow users to update the stored data without altering the QR code itself. These are commonly used in marketing campaigns, payment systems, and customer engagement platforms where the linked content needs frequent updates. Businesses use QR codes to enhance user experiences, such as enabling quick access to websites, digital menus, mobile payments, and loyalty programs.

In the field of **digital payments and authentication**, QR code generators play a crucial role. Mobile payment apps like Google Pay, Paytm, and Apple Pay rely on QR codes to facilitate seamless transactions between customers and merchants. Similarly, authentication mechanisms, such as two-factor authentication (2FA) and passwordless logins, often employ QR codes to verify user identities. For example, apps like WhatsApp Web and Google Authenticator generate QR codes that users can scan to securely link devices or verify accounts. The ability to quickly authenticate users while maintaining security makes QR codes an essential tool in digital ecosystems.

While QR codes offer convenience, they also come with potential security risks. **QR code phishing (also known as quashing)** is a growing threat where malicious QR codes redirect users to fraudulent websites that steal sensitive information. To mitigate these risks, QR code generators often incorporate encryption and validation mechanisms to ensure that the generated codes are safe. Some advanced QR code solutions also integrate **password protection** and **expiry settings** to limit unauthorized access. Users should always verify QR codes before scanning, especially in unfamiliar or untrusted environments.

The future of QR code generation is evolving with advancements in **Artificial Intelligence (AI), blockchain, and IoT (Internet of Things)**. AI-driven QR code generators can now analyze scanning patterns to optimize engagement strategies for businesses. Blockchain-based QR codes ensure **tamper-proof authentication**, enhancing their use in supply chain management and digital certifications. In IoT applications, QR codes are used to link physical devices with digital platforms, improving automation in smart homes, healthcare, and industrial monitoring. As industries continue to digitize processes, QR code technology will remain a fundamental tool for fast and efficient data exchange.

In conclusion, QR code generators have revolutionized the way information is shared, stored, and accessed. Whether for payments, marketing, authentication, or security, QR codes offer a fast and user-friendly solution for businesses and individuals alike. With continuous innovations and improvements, QR codes will continue to play a significant role in the digital transformation of various industries, enhancing convenience while addressing security concerns.

Key Parameters for Customizing a QR Code:

- **Data Type**: You can encode different types of data such as URLs, email addresses, text, phone numbers, or even Wi-Fi credentials.
- **Error Correction**: QR codes have built-in error correction, which allows the code to still be read even if part of it is damaged or obscured. There are different levels of error correction (L, M, Q, H), with H offering the highest error tolerance.
- **Size**: You can adjust the size (number of pixels) of the QR code depending on how large or small you want it.
- **Design Customization**: Some QR code generators offer customization options like changing the colors, adding logos, or altering the shapes of the code's elements.

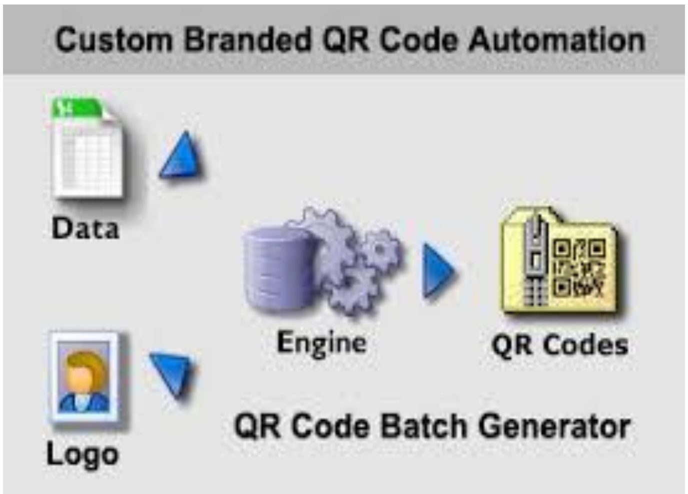

Fig 3.4 QR Code Generator

3.5 INTRODUCTION TO ARADUINO MICROCONTROLLER

The Arduino microcontroller is a versatile, open-source hardware platform used widely in various DIY electronics projects, robotics, automation systems, and embedded applications. It is a single-board microcontroller based on the ATmega328 or other similar chips, designed to be easy to use for both beginners and experienced developers. Arduino boards come with built-in Digital and Analog I/O pins, PWM (Pulse Width Modulation) support, and communication interfaces like UART, I2C, and SPI, making them highly flexible for interfacing with sensors, actuators, and other peripherals.

The Arduino microcontroller is also a compact and programmable device that serves as the brain for various electronic projects. It is widely used by hobbyists, students, and professionals for building embedded systems, Internet of Things (IoT) devices, robotics, and automation projects. The Arduino platform consists of both hardware (Arduino boards) and software (Arduino IDE), which makes it easy to write, upload, and execute programs. Unlike traditional microcontrollers, Arduino simplifies the development process by providing an open-source framework, making it accessible even for beginners.

An Arduino board is built around a microcontroller unit (MCU), which acts as the processing unit. Different Arduino models use different microcontrollers, such as the ATmega328P (Arduino Uno), ATmega2560 (Arduino Mega), or SAMD21 (Arduino Zero). The board also includes digital and analog input/output (I/O) pins, which allow it to interact with sensors, motors, LEDs, and other electronic components. Additionally, it features a USB interface for programming and power supply options, including battery and external adapters. Some advanced boards, like the Arduino Nano 33 IoT, come with built-in Wi-Fi and Bluetooth for wireless connectivity.

The **Arduino Integrated Development Environment (IDE)** is used to write, compile, and upload code to the microcontroller. The code, known as a **sketch**, is written in **C/C++**, using Arduino's simplified programming structure. The basic structure of an Arduino sketch consists of two main functions:

- setup(): Runs once when the board is powered on and is used to initialize settings.
- loop(): Executes continuously, controlling the behavior of connected components.

The Arduino platform also supports **third-party libraries**, which allow users to extend its functionality. These libraries enable interaction with devices like sensors, displays, motors, and communication modules (such as Wi-Fi and Bluetooth).

In the context of the Smart Box for Parcel Collection project, the Arduino microcontroller acts as the central control unit responsible for managing the entire system. The Arduino board interacts with various components, such as the RFID reader, QR code scanner, servo motors, communication modules, and the cloud database, to automate the parcel drop-off and collection process. It reads data from the RFID reader and QR code scanner, verifies parcel information, and controls the opening and closing of the compartments based on authentication.

Arduino microcontrollers are used in a wide range of applications across different fields:

1. **Home Automation** – Arduino is used to control lights, fans, and appliances through smart home systems.
2. **Robotics** – Many DIY robotic projects use Arduino to control motor movements, obstacle detection, and wireless communication.
3. **IoT (Internet of Things)** – With the integration of Wi-Fi or LoRa modules, Arduino enables the development of smart IoT devices like weather stations and security systems.
4. **Industrial Automation** – Arduino is used in industrial environments for monitoring temperature, controlling conveyor belts, and automating repetitive tasks.
5. **Wearable Technology** – Small Arduino boards like the **Arduino Nano** are used in wearable health devices and fitness trackers.

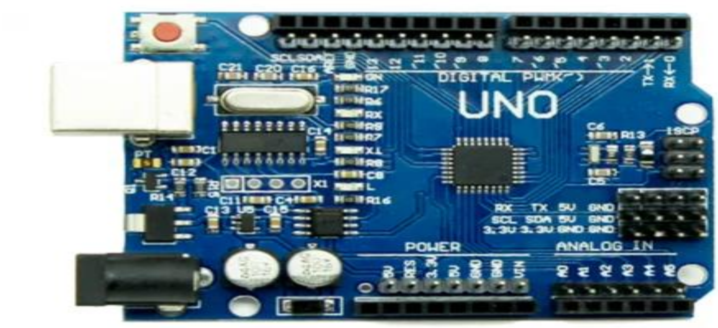

FIG 3.5 Arduino

3.6 QR CODE SCANNER

The **QR Code scanner** is a critical component for recipient authentication during parcel retrieval. When the recipient arrives at the Smart Box to pick up their parcel, they scan the unique QR code they received via email, SMS, or app. The scanner reads the QR code and sends the data to the microcontroller for verification. If the QR code is valid and corresponds to the parcel in the box, the system grants access to the designated compartment.

A QR Code Scanner is a device or application used to decode Quick Response (QR) codes, which are two-dimensional barcodes capable of storing information such as URLs, contact details, payment credentials, and product information. Unlike traditional barcode scanners that read only one-dimensional barcodes, QR code scanners can capture and interpret complex data in a fraction of a second. These scanners are widely used in various industries, including retail, healthcare, logistics, and digital transactions, enabling fast and contactless access to information.

A QR code scanner operates using **optical image recognition technology**. It captures an image of the QR code using a **camera or laser sensor** and then processes the data using a **QR code decoding algorithm**. The pattern of black and white squares in the QR code is analyzed to retrieve the encoded information. Most modern smartphones come with built-in QR code scanning capabilities through their native camera apps, while dedicated hardware scanners are used in commercial applications for higher accuracy and speed.

The process of scanning a QR code involves three main steps:

1. **Detection:** The scanner identifies the QR code within the camera's view.
2. **Decoding:** The software reads the unique arrangement of squares and extracts the embedded data.
3. **Action Execution:** Based on the decoded information, the scanner may open a website, process a payment, or display text.

The QR code scanner is often a camera-based or laser-based reader integrated into the Smart Box, ensuring fast, accurate, and secure verification.

QR code scanners come in different forms, depending on their functionality and usage:

1. **Mobile QR Code Scanners:** These are software-based scanners integrated into smartphones and tablets, commonly used for scanning website links, payments, and authentication.
2. **Handheld QR Code Scanners:** These are physical devices often used in retail stores and warehouses to scan product codes quickly.
3. **Fixed-Mount QR Code Scanners:** These are installed at entry points or kiosks for access control and automated check-ins.
4. **Online QR Code Scanners:** These web-based tools allow users to upload QR code images and decode the data instantly.

Despite their convenience, QR code scanners also pose security risks. **Malicious QR codes** can redirect users to phishing websites, download malware, or steal personal data. To mitigate these risks:

- Always use **trusted QR code scanning apps** that verify the authenticity of scanned codes.
- Check the **URL preview** before clicking on any link embedded in a QR code.
- Enable **security features like two-factor authentication (2FA)** when using QR codes for authentication or payments.
- Businesses should use **dynamic QR codes** with encryption and expiration settings to enhance security.

FIG 3.6 QR Code Scanner

3.7 COMPARTMENT CONTROL MECHANISM

The **compartment control mechanism** is responsible for physically unlocking and locking the Smart Box compartments. Once the recipient’s QR Code is authenticated, the microcontroller sends a signal to the compartment control system to open the appropriate compartment. This is typically achieved using **servo motors** or **solenoids**. **Servo motors** are used to control the precise movement of compartment doors, ensuring they unlock or lock as needed. **Solenoids** function as electromagnets that release or secure locks when energized. This mechanism provides a secure and reliable way to grant parcel access only to the correct recipient, preventing unauthorized access.

A Compartment Control Mechanism is a system used to regulate access, storage, and security within defined compartments or sections of a larger structure. This mechanism is commonly found in various applications, including parcel lockers, storage units, smart cabinets, and industrial automation systems. The fundamental purpose of a compartment control system is to ensure that only authorized users can access specific compartments while maintaining efficient organization and tracking of stored items. These systems are often integrated with RFID, QR code scanners, biometric authentication, or PIN-based entry systems to enhance security and usability.

Working Principle of Compartment Control Systems

The operation of a compartment control mechanism typically involves an **electronic locking system**, which can be activated through different authentication methods. When an authorized user attempts to access a compartment, the system verifies their credentials and unlocks the corresponding section. Many modern compartment control systems use **microcontrollers or embedded systems** to process authentication inputs and control the locking mechanism. Some systems also include **sensors and IoT connectivity**, allowing real-time monitoring of compartment status, occupancy levels, and security breaches.

For example, in a **smart parcel locker**, when a delivery person deposits a package, the system assigns it to a specific compartment. The recipient then receives a **QR code or OTP (One-Time Password)**, which they can use to retrieve the package securely. If an unauthorized person tries to open the locker, the system can trigger an alarm or notify administrators through a cloud-based dashboard.

3.8 DISPLAY UNIT

The **display unit** is an optional but helpful component that provides real-time information to users. It is typically an **LCD screen** or **LED panel** mounted on the Smart Box. This display can show the status of the system, including messages such as “Parcel Ready for Pickup,” “Scan QR Code to Access,” or “Invalid QR Code.” The display unit can also show error messages or system alerts, improving the user experience by providing clear instructions. Additionally, the display unit might provide information about the system’s status, such as which compartments are full or available, helping users navigate the system efficiently.

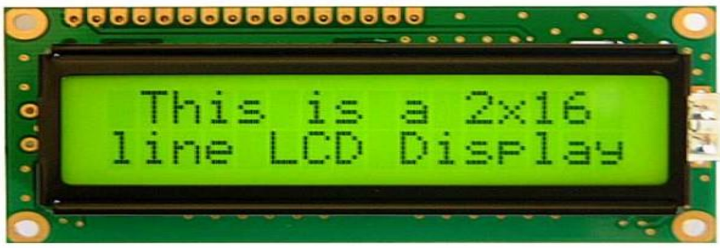

FIG 3.8 Lcd Display

3.9 CLOUD DATA BASE SERVER

The **cloud database** or **server** is the central repository for all parcel-related data in the Smart Box system. This database stores important information such as RFID tag IDs, QR code data, parcel statuses (e.g., “in transit,” “available for pickup,” or “picked up”), timestamps, recipient details, and system logs. The database ensures that all parcel transactions are tracked in real-time and can be accessed by both senders and recipients. The **cloud-based** solution enables remote access to parcel information and status updates, while also providing analytics for administrators to monitor system performance, usage patterns, and optimize the process.

A **cloud database server** refers to a database that is hosted and managed on a cloud computing platform rather than being installed on an on-premises server. It offers flexible, scalable, and cost-efficient database management. Cloud database servers are often used by organizations and individuals who want to store and manage data remotely with high availability, security, and scalability.

The buzzer can be triggered to produce different tones or patterns of sound, providing feedback during key moments of the parcel collection process. For instance, when a user successfully scans their QR code to authenticate their identity and unlock the compartment, the system can emit a short confirmation beep or series of beeps to signal success. This immediate auditory feedback assures the user that the system has processed their request successfully and the compartment is now accessible. Conversely, if the user attempts to access a compartment without proper authorization, such as by scanning an invalid QR code or using the wrong RFID tag, the system can trigger a warning tone or error beep to indicate an issue.

A Cloud Database Server is a remote, internet-based system used for storing, managing, and accessing databases without the need for physical infrastructure. Unlike traditional on-premises databases that require dedicated hardware and maintenance, cloud database servers operate on cloud computing platforms, offering scalability, flexibility, and high availability. These servers are hosted by cloud service providers such as Amazon Web Services (AWS), Google Cloud, Microsoft Azure, and Oracle Cloud, ensuring data is accessible from anywhere in the world.

A cloud database server functions as a centralized system that manages structured and unstructured data through database management software (DBMS). The data is stored across distributed cloud infrastructure, often using a multi-tenant architecture, where multiple users share the same physical resources while maintaining isolated environments. Cloud database servers operate in two main models:

1. **Managed Database Services:** The cloud provider handles all aspects, including database setup, maintenance, backups, and security. Examples include Amazon RDS, Google Cloud SQL, and Azure SQL Database.
2. **Self-Managed Database on Cloud Infrastructure:** Users deploy and manage their own database servers on cloud virtual machines. This model provides greater control but requires more administrative effort.

Cloud databases use automated scaling, load balancing, and replication to ensure optimal performance and reliability, allowing businesses to handle fluctuating workloads without service interruptions.

Types of Cloud Databases and Their Detailed Description

Cloud databases come in various types, each designed to handle different data structures, scalability requirements, and application needs. Below is an elaboration on the key types of cloud databases mentioned earlier.

1. Relational Cloud Databases (SQL-based)

Relational cloud databases follow the Structured Query Language (SQL) and store data in a structured format using tables, rows, and columns. They are ideal for applications requiring data integrity, consistency, and complex querying.

Features

- ACID Compliance (Atomicity, Consistency, Isolation, Durability) ensures reliable transactions.
- Schema-Based Structure means predefined tables and relationships between them.
- Strong Data Integrity and Normalization to reduce redundancy.

Use Cases

Enterprise applications – ERP, CRM, and financial systems.
E-commerce platforms – For managing customer orders, payments, and inventory.
Banking and Finance – Secure and structured transaction processing.

Examples of Relational Cloud Databases

- Amazon RDS (Relational Database Service) – Supports MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server.
- Google Cloud SQL – Fully managed SQL database service.
- Microsoft Azure SQL Database – Scalable and optimized for Microsoft-based applications.

2. NoSQL Cloud Databases (Non-Relational)

NoSQL (Not Only SQL) databases are designed to handle unstructured or semi-structured data such as JSON, XML, and key-value pairs. They are optimized for scalability, flexibility, and speed, making them suitable for Big Data and real-time applications.

Types of NoSQL Databases

1. Document-Based: Stores data in JSON, BSON, or XML format.
   - Example: MongoDB Atlas (Fully managed cloud MongoDB).
2. Key-Value Store: Simple, fast lookup of key-value pairs.
   - Example: Amazon DynamoDB, Redis, Couchbase.
3. Column-Family Store: Optimized for high-speed read/write operations, used in Big Data applications.
   - Example: Apache Cassandra, Google Bigtable.
4. Graph-Based: Best for relationship-heavy data like social networks and fraud.
   - Example: Neo4j Aura, Amazon Neptune.

Features

- High Scalability & Availability – Distributes data across multiple servers.
- Flexible Schema – No fixed table structure, allowing dynamic changes.
- Optimized for Speed – Handles high-throughput workloads.

Use Cases

Social Media Platforms – Storing user relationships and dynamic content.
IoT Applications – Handling time-series data from connected devices.
Real-Time Analytics – Processing logs and tracking user behavior.

3. NewSQL Cloud Databases

NewSQL databases combine the benefits of traditional SQL databases (ACID compliance, structured queries) with the scalability of NoSQL. These databases are ideal for applications requiring high-performance transactions and horizontal scalability.

Features

- Distributed Computing – Spreads workload across multiple cloud nodes.
- Strong Consistency – Ensures real-time data accuracy.
- Optimized for Cloud Environments – Designed to scale like NoSQL but with SQL capabilities.

Use Cases

High-Traffic Applications – Online gaming, real-time bidding.
Enterprise Applications – Large-scale financial and operational databases.
E-commerce and Digital Payments – Managing huge transactional volumes.

Examples of NewSQL Cloud Databases

- Google Spanner – Google's globally distributed SQL database.
- Cockroach DB – A cloud-native NewSQL database built for resilience and scalability.
- Mem SQL (Single Store) – Real-time analytics and data processing.

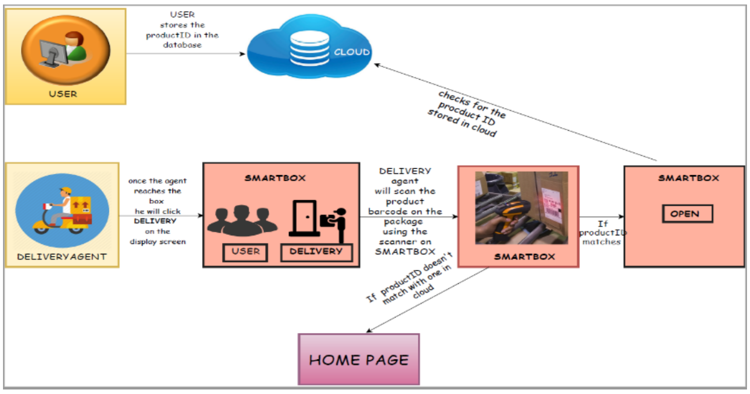

FIG: 3.9 Cloud Data Base Server

3.10 COMMUNICATION MODULE WIFI/BLUETOOTH

The communication module facilitates connectivity between the Smart Box and the cloud server, enabling real-time data transmission. This module can be based on Wi-Fi, Bluetooth, or Lora WAN, depending on the system's requirements and range. The communication module ensures that the system can update the cloud database, send notifications to recipients, and allow remote monitoring of the Smart Box's status. For example, the module could send a notification to the recipient when their parcel is ready for pickup, or update the cloud server when a parcel is scanned for pickup. Wi-Fi or Bluetooth connectivity allows the system to be integrated with mobile apps, providing a seamless user experience.

WIFI Communication Module

WIFI (Wireless Fidelity) is a wireless networking technology that allows devices to connect to the internet and communicate with each other over a local area network (LAN). It is based on the IEEE 802.11 standard and operates primarily on 2.4 GHz and 5 GHz frequency bands, with newer versions supporting the 6 GHz band for improved performance.

How WIFI Works

A WIFI communication module consists of a radio transceiver, antenna, and network protocol stack. It connects to a wireless access point (router) that enables communication between devices and the internet. WiFi uses two main architectures:

1. Infrastructure Mode – Devices connect through a central access point.
2. Ad-Hoc Mode (Peer-to-Peer) – Devices communicate directly without a router.

Features of WiFi Modules

High Data Transfer Speeds – Can reach up to 9.6 Gbps (WiFi 6).
Longer Range – Covers up to 100-200 meters indoors.
Supports Multiple Devices – Allows many devices to connect simultaneously.
Encryption & Security – Uses WPA3 for secure communication.
Internet Connectivity – Provides seamless access to online services.

Applications of WIFI Modules

- Smart Home Automation – WiFi is widely used in smart speakers, security cameras, and IoT devices.
- Industrial IoT (IoT) – Wireless control of machines, data monitoring, and cloud-based automation.
- Healthcare Devices – Remote patient monitoring, medical imaging, and smart health trackers.
- Mobile & Computing Devices – Laptops, tablets, and smartphones rely on WiFi for internet access.
- Streaming & Gaming – Provides low-latency and high-speed connectivity for video streaming and online gaming.

Popular WiFi Communication Modules

- ESP8266 & ESP32 – Low-cost, powerful microcontroller-based WiFi modules for IoT applications.
- Qualcomm QCA9377 – Used in laptops and embedded systems for WiFi connectivity.
- Realtek RTL8188 – Commonly used in USB WiFi adapters.

Bluetooth Communication Module

Bluetooth is a short-range wireless communication technology used for exchanging data between devices over short distances. It operates on the 2.4 GHz ISM (Industrial, Scientific, and Medical) band and follows the IEEE 802.15.1 standard. Bluetooth modules consume less power compared to WiFi, making them ideal for low-power IoT and wearable devices.

How Bluetooth Works

Bluetooth modules create a Personal Area Network (PAN) by establishing a connection between paired devices. It operates in different modes:

1. Classic Bluetooth (BR/EDR) – Used for continuous data streaming (e.g., audio devices).
2. Bluetooth Low Energy (BLE) – Optimized for low power, suitable for IoT applications.
3. Bluetooth Mesh – Allows many-to-many communication, ideal for smart lighting and industrial automation.

Features of Bluetooth Modules

Low Power Consumption – BLE modules can operate for years on a small battery. Short-Range Communication – Typically 10–100 meters depending on power class. Secure Pairing & Encryption – Uses AES-128 encryption for secure connections. Easy Pairing – Quick and simple device discovery and connection. Supports Multiple Devices – Can connect to multiple peripherals simultaneously.

Applications of Bluetooth Modules

- Wearable Technology – Smartwatches, fitness trackers, and medical devices use Bluetooth for data transfer.
- Wireless Audio – Bluetooth speakers, headphones, and car audio systems.
- Home Automation – Smart locks, lighting systems, and remote-controlled appliances.
- Automobile Industry – Hands-free calling, Bluetooth-enabled infotainment systems.

- Gaming Controllers & Peripherals – Used in wireless keyboards, mice, and gaming controllers.

Popular Bluetooth Communication Modules

- HC-05 & HC-06 – Classic Bluetooth modules for serial communication.
- NRF52840 – BLE module with advanced security and low power consumption.
- CSR8675 – High-quality audio Bluetooth module for wireless headsets.

A **communication module** for **Wi-Fi** or **Bluetooth** is a hardware component that enables wireless communication between devices. These modules are widely used in embedded systems, IoT (Internet of Things) devices, and mobile applications to provide connectivity for transmitting and receiving data. Wi-Fi and Bluetooth modules are essential for applications like smart home systems, wearables, health monitoring devices, and industrial automation.

FIG: 3.10 Communication Module Wifi/Bluetooth

3.11 POWER SUPPLY

A power supply is a crucial element in any electrical or electronic system, providing the necessary energy to operate various devices. It serves as the interface between the power source and the equipment, ensuring that the electrical energy is converted into a form suitable for use. Power supplies are used in a vast range of applications, from household electronics and industrial machines to communication systems and medical devices. Without a stable power supply, electronic components may malfunction, leading to performance issues or even damage.

Power supplies can be broadly classified into AC to DC power supplies, DC to DC power supplies, linear power supplies, switching mode power supplies (SMPS), uninterruptible power supplies (UPS), and battery power supplies. AC to DC power supplies convert alternating current (AC) from mains electricity into direct current (DC), which is needed for most electronic circuits. DC to DC power supplies, or converters, modify the voltage level of a DC source to meet specific requirements. These are commonly found in portable gadgets and electric vehicles.

A linear power supply provides a steady DC output by using a transformer, rectifier, and voltage regulator. While it offers high stability and low noise, it is less efficient due to significant heat dissipation. In contrast, a Switching Mode Power Supply (SMPS) is far more efficient, as it converts power using high-frequency switching regulators. SMPS units are commonly used in computers, televisions, and industrial automation systems, offering lighter weight, higher efficiency, and minimal heat generation.

For critical applications where power interruptions can lead to data loss or damage, Uninterruptible Power Supplies (UPS) are used. A UPS provides temporary backup power during outages, ensuring that systems like computers, data centers, and hospital equipment continue functioning until an alternate power source takes over. Similarly, battery power supplies serve as a portable energy source, converting chemical energy into electrical energy to power mobile devices, remote sensors, and emergency lighting systems.

A power supply consists of several key components, including a transformer (to step up or stepdown voltage), a rectifier (to convert AC to DC), a filter (to smoothen the DC signal), and a voltage regulator (to maintain a stable output voltage). In high-power applications, heat sinks and cooling fans are often added to dissipate excess heat, preventing overheating and ensuring safe operation.

Higher efficiency reduces energy waste, while low ripple and noise are essential for sensitive electronic circuits. Power supplies are integral to computers, industrial automation, telecommunication systems, medical equipment, and renewable energy systems, making them one of the most important components in modern electrical and electronic engineering. As technology advances, power supply designs continue to evolve, focusing on energy efficiency, miniaturization, and improved performance.

Types of Power Supplies

Power supplies are classified into various types based on their operation and functionality. Some of the most common types include

1. AC to DC Power Supply

Most electronic devices operate on direct current (DC), while the electricity supplied from the power grid is alternating current (AC). An AC to DC power supply converts the incoming AC voltage into a stable DC voltage suitable for electronic circuits. This process typically involves rectification, filtering, and voltage regulation to ensure smooth DC output. Example: Laptop chargers, mobile phone adapters, desktop power supplies.

2. DC to DC Power Supply

A DC-to-DC power supply takes a DC input voltage and converts it into a different DC output voltage. These converters are used in applications where different components require different voltage levels, such as portable devices, automobiles, and embedded systems.

3. Linear Power Supply

A linear power supply operates by using a transformer, rectifier, and voltage regulator to produce a stable DC voltage. While these power supplies provide low-noise output, they are less efficient because a significant portion of energy is dissipated as heat.
Example: Audio equipment, laboratory power supplies, and precision analog circuits.

4. Switching Mode Power Supply (SMPS)

Unlike linear power supplies, SMPS uses high-frequency switching regulators to efficiently convert electrical power. This reduces power loss, making SMPS more compact and energy-efficient. However, due to the switching process, SMPS may introduce electrical noise into the circuit.
Example: Power adapters for computers, televisions, and industrial automation systems.

5. Uninterruptible Power Supply (UPS)

A UPS (Uninterruptible Power Supply) ensures continuous power during electrical failures by using a battery backup system. When the primary power source fails, the UPS instantly switches to battery power, preventing downtime and potential data loss.
Example: Backup systems for servers, hospitals, and emergency lighting.

6. Battery Power Supply

A battery power supply stores chemical energy and converts it into electrical energy. Batteries can be primary (non-rechargeable) or secondary (rechargeable) and are widely used in portable electronics, remote control devices, and electric vehicles.
Example: Lithium-ion batteries in smartphones and lead-acid batteries in cars.

7. Renewable Energy Power Supply

With the increasing demand for sustainable energy, renewable power supplies convert energy from solar panels, wind turbines, or fuel cells into usable electricity. These power sources require inverters and charge controllers to regulate voltage and store energy efficiently.
Example: Solar power systems, wind energy converters, hydrogen fuel cells.

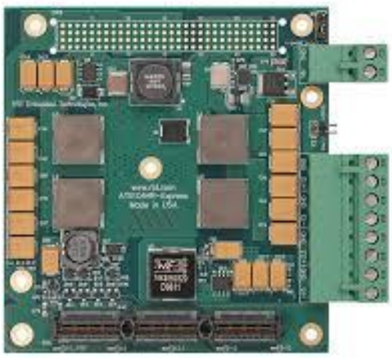

FIG: 3.11 Power Supply

3.12 USER INTERFACE

The **user interface** (either mobile app or web portal) serves as the interface for both recipients and administrators to interact with the Smart Box system. The recipient can use the app or portal to track their parcel in real-time, receive notifications, and manage the parcel pickup process. The app might also include features like QR code scanning for authentication and status updates on the system's availability. For administrators, the web portal offers tools to monitor the system's health, check for any issues or maintenance needs, and manage the cloud database. This user-friendly interface ensures seamless interaction between users and the Smart Box.

The Smart Box for Parcel Collection is designed to provide a secure and automated system for receiving packages using Arduino, RFID, and QR code technologies. The user interface (UI) plays a crucial role in guiding both delivery personnel and recipients through the process of dropping off and collecting parcels efficiently. The interface consists of a display screen, input methods such as RFID and QR scanners, status indicators, and security alerts.

The display panel is the central component of the UI. It can be a 16x2 LCD screen for basic systems or a touchscreen display for more advanced implementations. The screen provides important messages such as "Welcome to Smart Parcel Box," instructions for users, and parcel status updates. It also prompts users for authentication by displaying messages like “Scan RFID / QR Code to Access” and “Parcel Stored Successfully” upon successful delivery. If an error occurs, such as an incorrect QR code or unauthorized RFID tag, the display shows an error message to alert the user.

To ensure secure access, the Smart Box features multiple input methods. The RFID scanner allows delivery personnel or recipients to authenticate themselves using an RFID card, unlocking the designated compartment for placing or retrieving a package. Alternatively, the QR code scanner provides another authentication method where recipients can scan a unique QR code received via SMS or email. In some versions of the system, a numeric keypad may be included to allow recipients to enter a one-time PIN (OTP) for added security.

The status indicators further enhance the user experience by providing visual and audio feedback. LED indicators display different system states: a green LED signals that the box is available for a new delivery, a red LED indicates that the box is occupied, and a blinking red LED warns of an incorrect authentication attempt. Additionally, the buzzer system emits different beeps based on system events. A short beep confirms successful authentication, a long beep alerts the user to an error, and a continuous beep sound if the compartment door remains open for too long.

The user flow of the Smart Box is designed for simplicity and efficiency. When a delivery personnel approaches, they scan their RFID card or enter credentials. If authenticated, the compartment unlocks, allowing them to place the package inside. The system then sends a notification to the recipient about the package arrival. When the recipient arrives, they scan their RFID card or QR code, unlocking the compartment for retrieval. Once the parcel is collected, the system updates its status to "Available" for the next delivery.

Beyond basic functions, the Smart Box can include advanced features to enhance security and convenience. A mobile app or web dashboard can allow users to monitor parcel status remotely, receive real-time notifications, and even generate new access credentials if needed. A time-based locking system can be implemented to send reminders if a package is not picked up within a certain period. Additionally, camera integration can capture images of delivery personnel and recipients to ensure further security and verification.

The Smart Box UI combines clear instructions, intuitive input methods, and responsive feedback mechanisms to create a user-friendly and secure parcel collection system. By integrating Arduino with RFID, QR codes, LED indicators, and mobile connectivity, the Smart Box simplifies package management, making it an ideal solution for residential buildings, offices, and e-commerce deliveries.

FIG: 3.12 User Interface

3.13 CAMERA

The integration of a camera module in a Smart Box for parcel collection significantly enhances security by capturing images or videos during package drop-off and retrieval. This feature helps in preventing theft, verifying deliveries, and ensuring secure access. By incorporating a camera, the system can record visual proof of who interacts with the parcel box, making it an essential addition to automated parcel management solutions. The camera can be programmed to capture snapshots of delivery personnel and recipients, ensuring that only authorized individuals access the box.

Selecting the right camera module is crucial for smooth operation. Some commonly used modules for Arduino-based systems include the OV7670 camera, which is a simple and cost-effective option for capturing images, and the ESP32-CAM module, which supports Wi-Fi connectivity for remote monitoring and image transmission. For more advanced implementations, a Raspberry Pi camera module can be used to provide high-resolution images and support for AI-based facial recognition. Depending on the level of security needed, different camera modules can be chosen based on resolution, storage options, and connectivity features.

The working mechanism of the camera module is automated to activate under specific conditions. When delivery personnel scans their RFID card or QR code, the camera immediately captures an image and stores it for proof of delivery. Similarly, when a recipient scans their authentication method, the system records their image, ensuring the right person collects the package. If an unauthorized attempt is detected, the camera takes a snapshot, stores the image, and can even send a notification to the owner. Additionally, tamper detection mode ensures that if someone tries to force open the box, the camera records the activity and alerts the system.

Adding a camera module offers multiple benefits, making the Smart Box more reliable and secure. The camera provides real-time monitoring if connected to the internet, allowing users to check their parcel status remotely. Advanced versions can include face recognition technology, enabling contactless authentication for package retrieval. In case of disputes over lost or damaged parcels, the system provides time-stamped proof of delivery, helping resolve issues efficiently. Moreover, the camera can be integrated with motion detection sensors, ensuring that images are only captured when movement is detected, thereby saving storage space and energy.

For seamless integration, the camera module can be connected to an Arduino board, especially when using ESP32-CAM, which enables wireless image transmission. The captured images or videos can be stored in an SD card, uploaded to a cloud database, or emailed to the user for verification. The system can also be linked to IoT platforms like Google Drive, Firebase, or AWS, allowing users to access live footage of their parcel box. Additionally, the camera can be synchronized with buzzer alerts and LED indicators, triggering an alarm if unauthorized access is detected.

In conclusion, incorporating a camera module into a Smart Box for parcel collection enhances security, improves user confidence, and prevents unauthorized access. Whether using a basic image-capturing module or an AI-powered real-time monitoring system, integrating a camera ensures safe and efficient parcel management. Future advancements could include AI-based object detection, night vision cameras, and cloud-based AI processing, making the system even smarter and more robust.

3.14 Servo Motors in the Smart Box for Parcel Collection Project

A servo motor is a specialized type of DC motor designed for precise control of angular position, speed, and acceleration. Unlike regular DC motors, which rotate continuously, servo motors are designed to rotate to a specific position within a defined range (typically 0° to 180° or 360° depending on the servo). They are essential for applications requiring accurate movement, positioning, and torque control, such as in robotic arms, camera pans, and automated systems like the Smart Box for Parcel Collection.

In this project, servo motors are primarily used to control the locking mechanism of the compartments where parcels are stored. When a recipient scans their QR Code for parcel retrieval, the system must securely unlock the corresponding compartment to allow access. Here's how servo motors contribute to this process:

Working Principle of Servo Motors

A servo motor consists of:

1. **DC Motor**: Provides rotational motion.
2. **Control Circuitry**: Receives signals (PWM) from a microcontroller to set the desired angle of rotation.
3. **Feedback System (Potentiometer)**: Ensures the motor moves to the correct position by providing feedback to the controller.

The **microcontroller** (e.g., Arduino) sends a Pulse Width Modulation (PWM) signal to the servo. The PWM signal specifies the desired position, and the servo motor rotates to that specific angle. The servo motors used in this project can achieve precise rotational control, which is necessary for unlocking or locking the compartments.

FIG: 3.14 Servo Motor

5.15 The Real-Time Clock (RTC)

An RTC module typically contains the following key components:

- **Quartz Crystal Oscillator**: Provides the clock signal that drives the timekeeping functionality.
- **Time Registers**: Stores the current date and time in a digital format.
- **Battery Backup**: A small coin cell battery (usually CR2032) ensures the clock keeps running even when the main system is powered off or disconnected from a power source.

The RTC module communicates with the **microcontroller** (e.g., Arduino) via standard communication protocols such as **I2C** or **SPI**, which allows the microcontroller to read and write time data to the RTC module. The system uses this data for logging actions and syncing time-sensitive operations.

In the **Smart Box for Parcel Collection**, the **RTC module** offers several key benefits that enhance the functionality and security of the system:

1. **Accurate Time Stamping:**

- The RTC module ensures that every **parcel transaction** (drop-off, pickup, etc.) is **timestamped accurately**. This is essential for tracking when parcels are placed in the system, when they are picked up, and for generating logs that can be reviewed later. For example, the system can log:
  - The exact time when a parcel is dropped off.
  - The time when a recipient collects their parcel after scanning the QR code.
  - Time-related actions for **maintenance** or **system checks**.

This timestamping can be used for reporting, audit purposes, or simply to verify that the system is functioning correctly.

2. Tracking Time-Based Events:

- Time-based events like **notification reminders** or **reminder alerts** for parcel pickups can be integrated with the RTC. For example:
  - If a parcel has not been picked up within a specific time frame, the system could send a reminder message or trigger an alert to the recipient.

- The system can automatically reset compartments or trigger alarms after a certain period, ensuring that parcels that remain in the box for too long are flagged for review.

This functionality ensures that parcels do not remain in the system for an extended period, enhancing the efficiency of the service.

2. **Maintaining System Time During Power Interruptions:**

   One of the most important features of an RTC module is its **ability to keep time even when the power is lost**. When the system is powered off or rebooted, the RTC module continues to run on the backup battery, ensuring that the time is maintained without any interruption. This eliminates the need for the system to rely on external time sources (such as network time) and avoids issues related to power failures or resets.

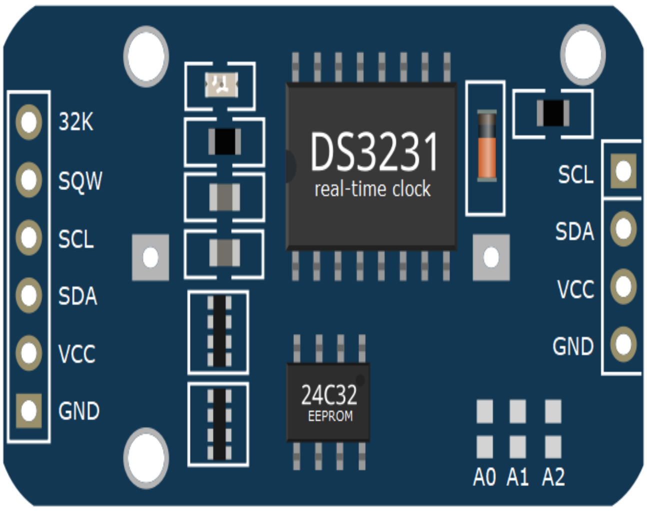

FIG 3.15 Real Time Clock Module With Arduino

5. Calendar Functions:

- Many RTCs can handle **leap years**, **months with varying days**, and **daylight-saving time** adjustments. This allows the RTC to track not just hours and minutes, but also months, years, and weekdays accurately.

How Real-Time Clock Works:

- The RTC contains a **crystal oscillator** (often a 32.768 kHz crystal) that keeps track of time by generating a stable frequency.

- The RTC keeps count of the seconds, minutes, hours, and so on, using counters that increment with each cycle of the crystal.
- When the system is powered off, the battery keeps the crystal and the time-keeping circuit running, allowing the RTC to continue counting time.

3.16 BUZZER

In the Smart Box for Parcel Collection project, the buzzer plays a critical role in providing audible feedback and enhancing the user interaction with the system. As an essential component for user communication, the buzzer alerts the user to various system statuses and events, making it a key part of the user experience and security features.

The buzzer can be triggered to produce different tones or patterns of sound, providing feedback during key moments of the parcel collection process. For instance, when a user successfully scans their QR code to authenticate their identity and unlock the compartment, the system can emit a short confirmation beep or series of beeps to signal success. This immediate auditory feedback assures the user that the system has processed their request =successfully and the compartment is now accessible.

The buzzer also plays a role in enhancing the security of the system. If a tamper detection sensor (like a shock or vibration sensor) is triggered, indicating unauthorized access attempts or tampering with the system, the buzzer can emit a loud, attention-grabbing sound, deterring potential tampering and alerting nearby personnel or users of suspicious.

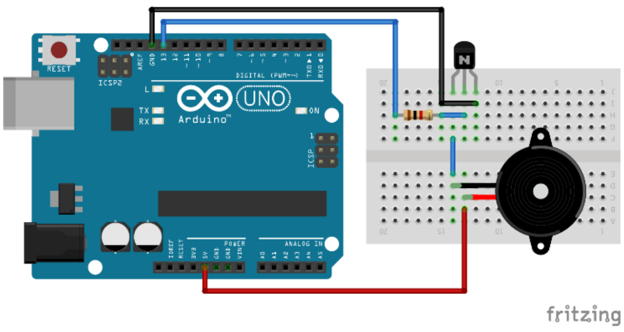

FIG: 3.16 Buzzer

CHAPTER 4

SOFTWARE REQUIREMENTS

4.1 ARDUINO SOFTWARE

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos are built around an ATmega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output.

What you will need

- A computer (Windows, Mac, or Linux)
- An Arduino-compatible microcontroller (anything from this guide should work)
- A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer
- A Raspberry pi Pico
- Windows 7, Vista, and XP
- Installing the Drivers for the Raspberry pi

Software

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Platform

A computing platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries.

Operating system is one of the requirements mentioned when defining system requirements. Software may not be compatible with different versions of the same line of operating systems, although some measure of backward compatibility is often maintained.

For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernelv2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v.

APIs And Drivers

Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms

How to Download Arduino IDE

- You can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.
- You can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.
- If you aim to download Windows app version, make sure you have Windows 8.1 or Windows 10, as app version is not compatible with Windows 7 or older version of this operating system.

- You can download the latest version of Arduino IDE for Windows (Non-Admin standalone version)

The IDE environment is mainly distributed into three sections

1. Menu Bar
2. Text Editor

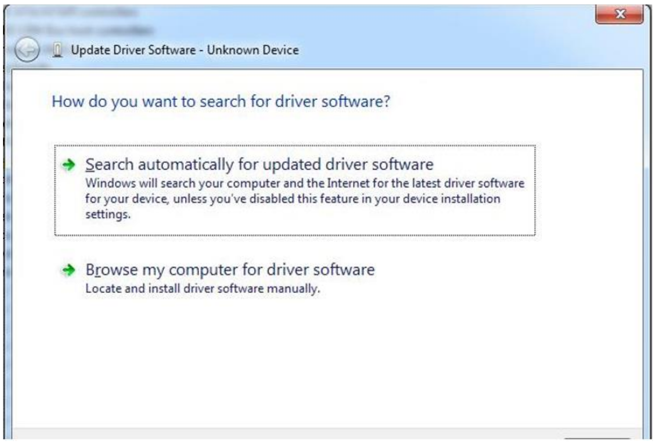

Fig:4.1 Driver Selection

- Plug in your board and wait for Windows to begin its driver installation process After a few moments, the process will fail, despite its best efforts
- Click on the Start Menu, and open up the Control Panel
- While in the Control Panel, navigate to System and Security. Next, click on System Once the System window is up, open the Device Manager
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COM)".
- If there is no COM & LPT section, look under 'Other Devices' for 'Unknown Device'
- Right click on the "Arduino UNO (COM)" or "Unknown Device" port and choose the "Update Driver Software" option Next, choose the "Browse my computer for Driver software" option.

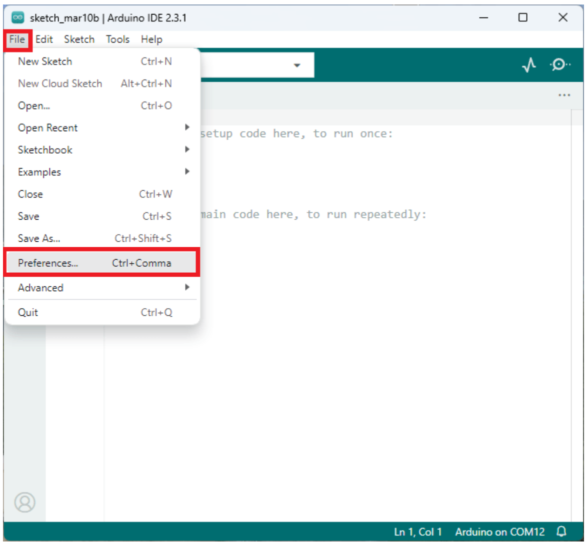

Fig:4.2 Preference Selection

2. Enter the following URL into the "Additional Boards Manager URLs" field: https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

Then, click the "OK" button:

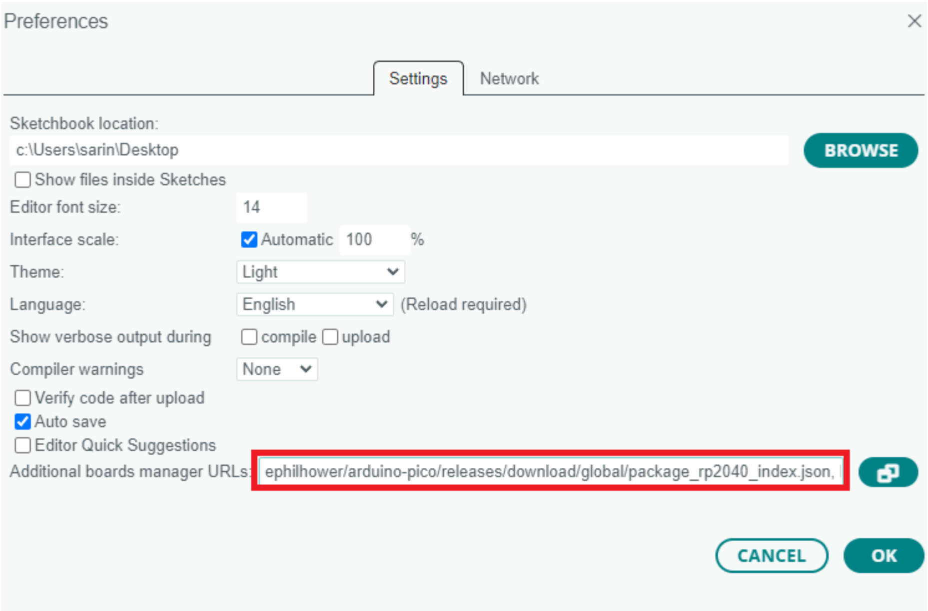

Fig:4.3 Board Manager

3. Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager…**

4. Search for "**Pico**" and install the Raspberry Pi Pico/RP2040 boards.

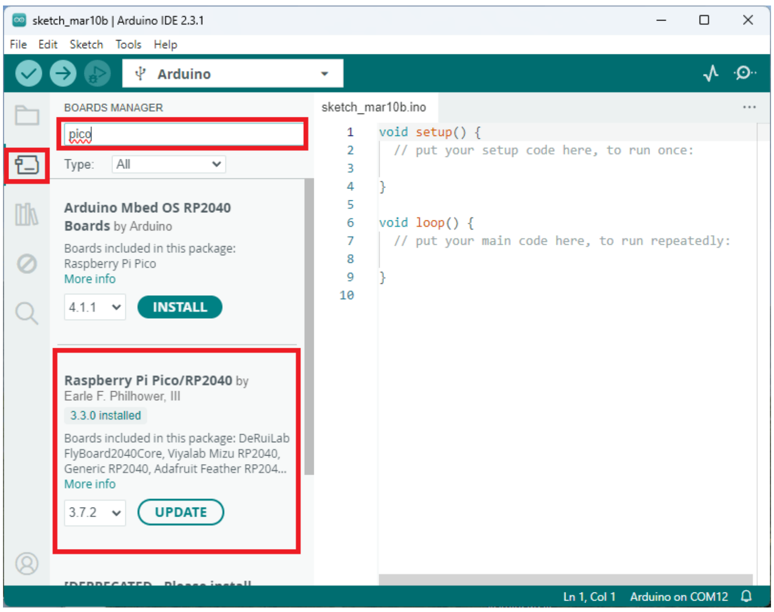

Fig:4.4 Installing Packages

5. That's it. It will install after a few seconds.

6. Now, if you go to **Tools** > **Board**, there should be a selection of Raspberry Pi Pico boards.

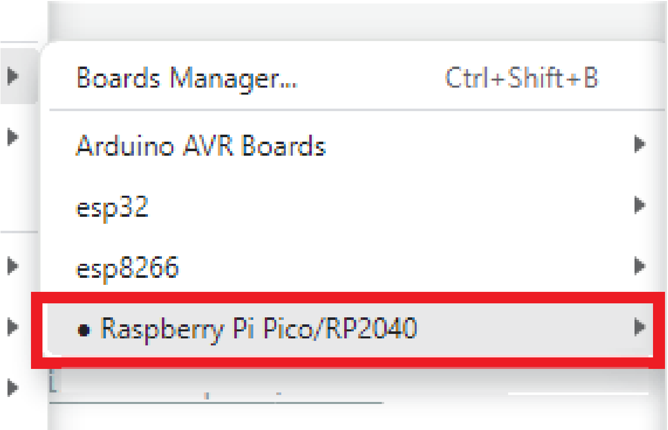

Fig:4.5 Selecting Board Model

9. **Connect your Raspberry Pi Pico** and using Device Manager **locate the COM port** that it is connected to.

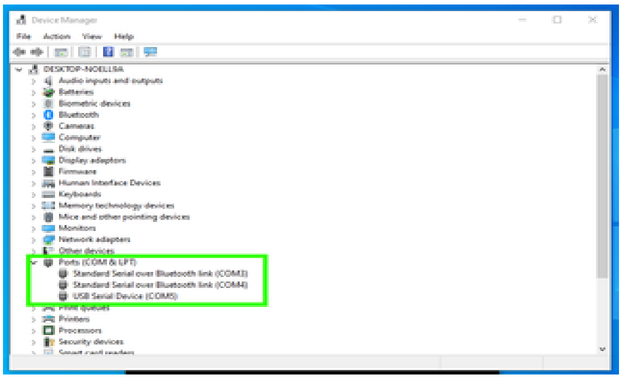

Fig:4.6 Selecting Port

10. Under Tools >> Port, **set the COM port** for the Raspberry Pi Pico.

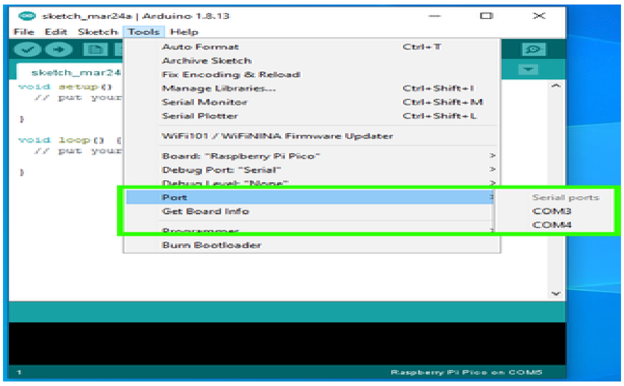

Fig:4.7 Selecting Com

11. **Open Files >> Examples >> Basics >> Blink** to test that we can write code to the Arduino.

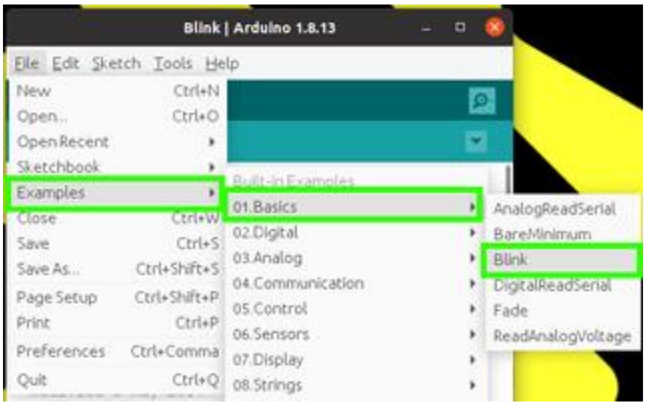

Fig:4.8 Example Codes

12. **Click on Upload** to write the code to the Raspberry Pi Pico. The default Blink sketch will flash the green LED next to the micro-USB port on the Raspberry Pi Pico

CHAPTER 5

WORKING MODEL AND COMPONENTS

5.1 Working Model

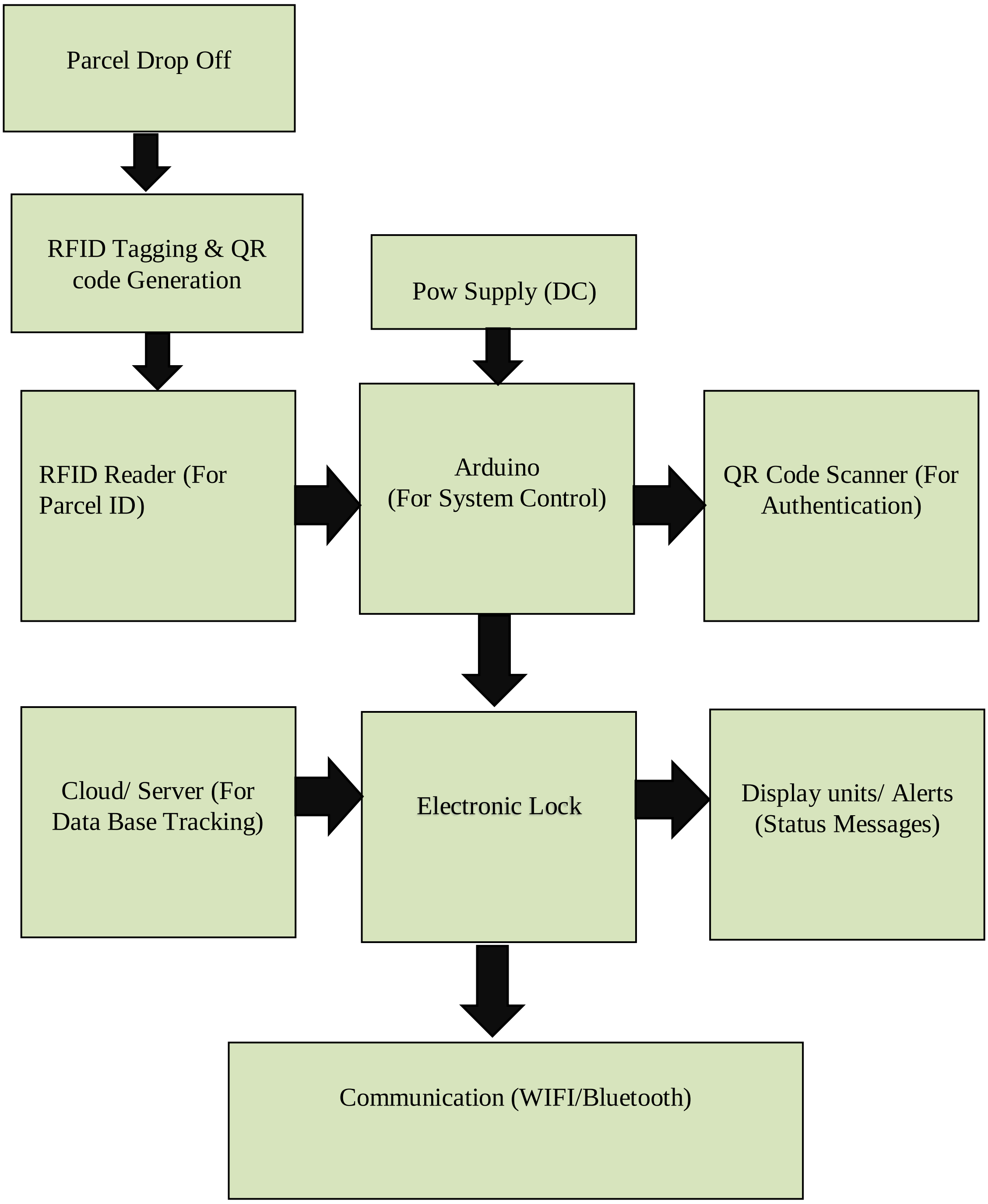

Fig: 5.1 Block Diagram of Smart Box For Parcel Collection

5.2 WORKING

The Smart Box for Parcel Collection using RFID and QR Code operates as an automated, secure system that facilitates both parcel drop-off and retrieval. When a sender wants to deliver a parcel, the system requires the parcel to be tagged with an RFID tag, which contains a unique identifier. Once the parcel is tagged, the sender places it in one of the compartments of the Smart Box. The RFID reader in the Smart Box scans the parcel's RFID tag, capturing the unique ID and registering the parcel in the system. The system then generates a unique QR Code linked to the parcel, containing important tracking and pickup details. This QR code is sent to the recipient via email, SMS, or a mobile app, notifying them that the parcel is ready for collection.

When the recipient arrives at the Smart Box to pick up their parcel, they scan the QR Code using their smartphone or the QR code scanner integrated into the box. The system authenticates the code and verifies that it corresponds to a parcel placed in the box. If the QR code is valid, the microcontroller sends a signal to unlock the corresponding compartment using servo motors or solenoids, allowing the recipient to retrieve their parcel. The system is designed to provide real-time tracking through the use of RFID technology, continuously monitoring the parcel's location and status. Notifications are sent to the recipient at every step of the process, from parcel arrival to pick up.

The system also ensures security by allowing only authorized recipients to access their parcels, as the QR Code serves as a secure authentication method. If an invalid or unauthorized code is scanned, the system will prevent access and notify the user. All parcel-related data, such as RFID IDs, pickup status, timestamps, and recipient details, are stored in a cloud database for real-time tracking and management. This database also facilitates notifications and updates, ensuring transparency and user convenience. Overall, the system integrates RFID and QR Code technologies to automate the parcel collection process, making it more efficient, secure, and user-friendly, while providing 24/7 accessibility for recipients.

The system authenticates the code and verifies that it corresponds to a parcel placed in the box. If the QR code is valid, the microcontroller sends a signal to unlock the corresponding compartment using servo motors or solenoids, allowing the recipient to retrieve their parcel. The system is designed to provide real-time tracking through the use of

RFID technology, continuously monitoring the parcel's location and status. Notifications are sent to the recipient at every step of the process, from parcel arrival to pick up.

CHAPTER 6

RESULTS

RESULTS

The Smart Box for Parcel Collection using RFID Reader and QR Code with Arduino successfully authenticates users by allowing access only to those with registered RFID cards or QR codes. It ensures secure parcel storage and retrieval through an automated locking mechanism that keeps the box locked until an authorized user access it. The system demonstrates high accuracy in RFID detection and QR code scanning, with minimal response time for authentication, making the process efficient. It is integrated with a camera, users can see the real time updates with the help of a IP address which shows the real time actions.

In terms of hardware performance, the Arduino microcontroller efficiently processes multiple authentication requests without lag, while the system is optimized for low power consumption, making it suitable for battery-powered operation. Security is enhanced by preventing unauthorized access, and the system maintains reliable performance under different environmental conditions.

Overall, the Smart Box offers a secure, automated, and efficient parcel collection system, reducing manual handling and improving convenience for users.

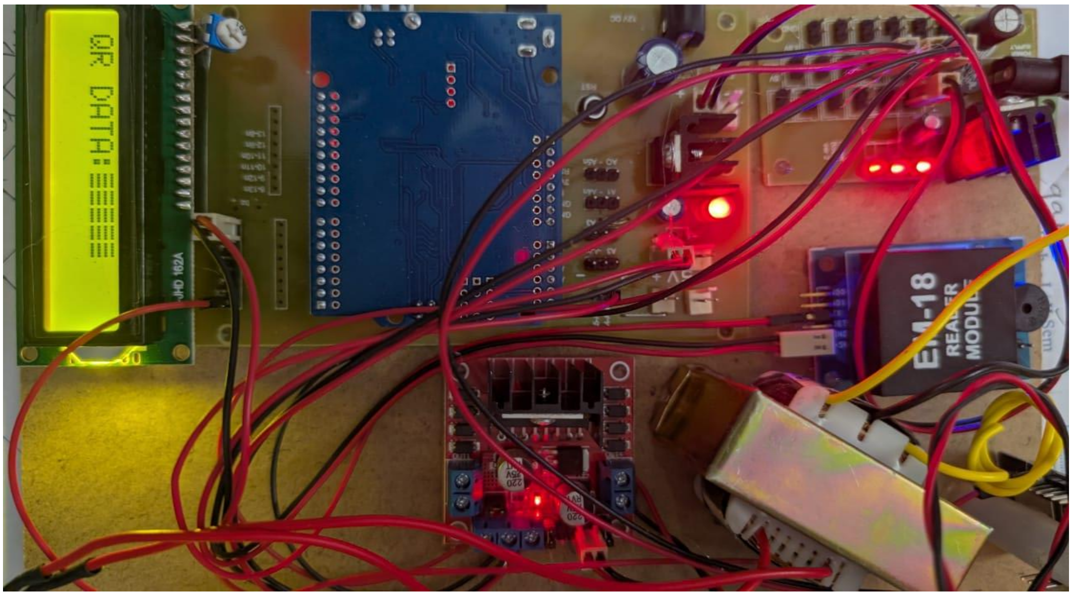

Fig: Smart Box for Parcel Collection

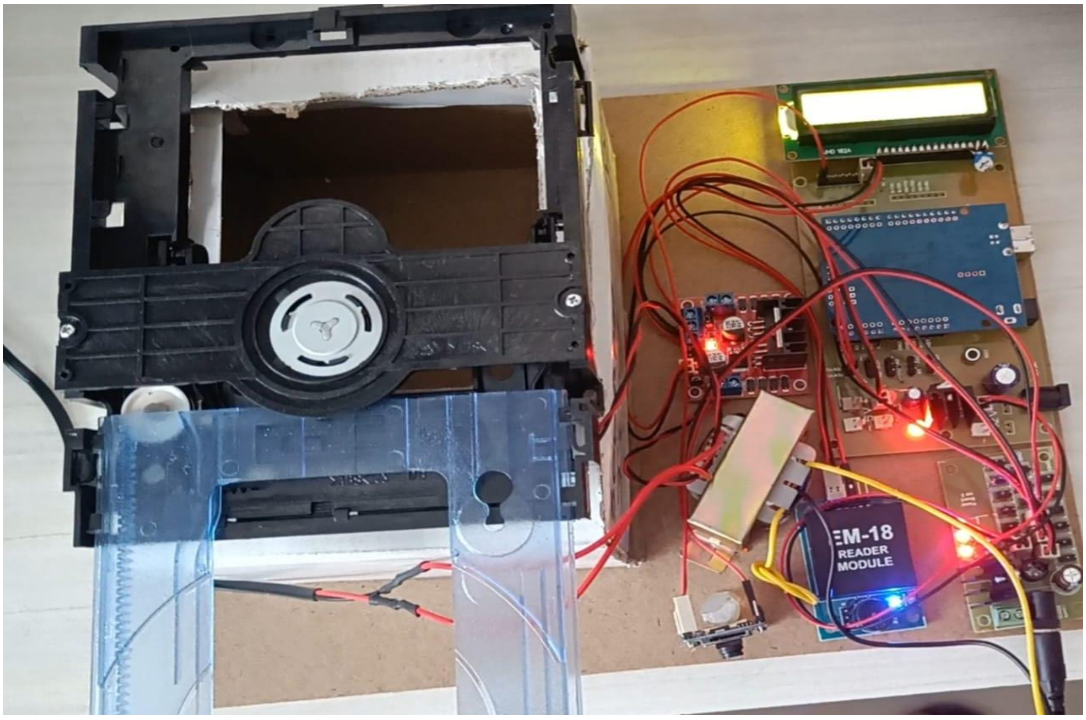

Fig: Result1

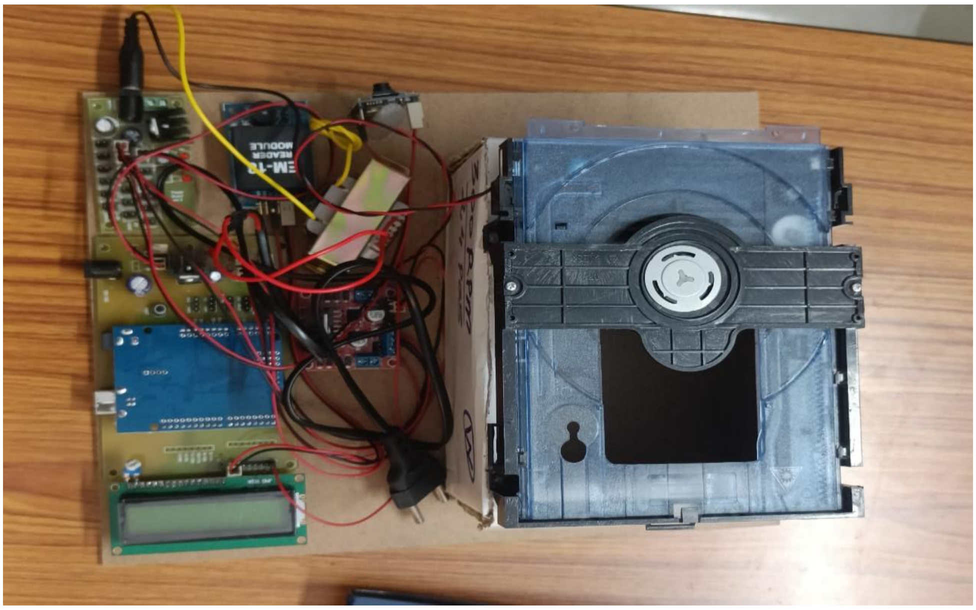

Fig: Result2

Advantages of The Smart Box for Parcel Collection

Key Benefits of the Smart Box for Parcel Collection

1. Convenience & Accessibility – Operates 24/7, allowing users to collect parcels anytime. QR code/RFID-based access ensures a quick, contactless pickup without staff involvement.
2. Enhanced Security – QR code/RFID authentication ensures only authorized access. Tamper detection prevents unauthorized access, while time-stamped logs track parcel activities.
3. Automated Parcel Management – Reduces human intervention and errors, assigns compartments based on parcel size, and streamlines storage space usage.
4. Cost-Effective Solution – Low operational costs, minimal maintenance, and affordable Arduino-based components make it budget-friendly.
5. Improved User Experience – Quick access, buzzer feedback, and instant notifications enhance user convenience and engagement.
6. Time Efficiency – Fast QR/RFID scanning enables instant retrieval, while time-stamped logs improve tracking and organization.
7. Scalability & Flexibility – Easily expandable to accommodate higher parcel volumes, with customizable compartments for different package sizes.
8. Environmental Benefits – Reduces delivery vehicle trips, lowering carbon footprint, and uses low-power components for energy efficiency.
9. Minimal Human Error – Automated sorting and retrieval eliminate misplacement and delivery errors, improving accuracy and efficiency in parcel management.

APPLICATIONS

Applications of Smart Box for Parcel Collection

A Smart Box for parcel collection is an automated, secure storage system designed to receive, store, and distribute parcels efficiently. Equipped with RFID, QR code authentication, IoT connectivity, and security features like cameras, these smart boxes are used across various industries to ensure secure and contactless package handling. Below are some of the key applications of smart parcel boxes:

1. **Residential Complexes** – Secure parcel storage for residents, preventing theft and missed deliveries.
2. **Corporate Offices** – Safe handling of business documents and employee packages

with ID-based access.

3. **E-Commerce & Logistics** – Automated last-mile delivery, reducing failed deliveries and costs.
4. **Universities & Colleges** – Secure storage for books, study materials, and exam documents.
5. **Hotels & Hospitality** – Guest deliveries, luggage storage, and room service automation.
6. **Retail & Shopping Centers** – Click-and-collect lockers for online orders, reducing waiting times.
7. **Hospitals & Healthcare** – Secure medicine distribution and document storage for patients and staff.
8. **Smart Cities** – Public parcel lockers in transport hubs for convenient urban deliveries.
9. **Government & Legal Sector** – Secure distribution of confidential documents and identity materials.
10. **Industrial & Warehouses** – RFID-based storage for tools and machine parts, improving efficiency.
11. **Rural Areas** – Reliable parcel collection points bridging the gap between e-commerce and remote locations.
12. **Airports & Travel Industry** – Secure lost-and-found storage and retrieval for passengers.

CHAPTER 7
CONCLUSION

Conclusion

The Smart Box for Parcel Collection using RFID Reader and QR Code with Arduino provides a secure, automated, and efficient solution for parcel management. By utilizing RFID and QR code authentication, the system ensures that only authorized users can access their parcels, reducing the chances of unauthorized access or misplaced deliveries. The automated locking mechanism enhances security, while real-time notifications improve user convenience.

The system demonstrates high accuracy, quick response time, and low power consumption, making it a reliable and practical solution. Although minor challenges such as RFID misalignment, QR code scanning under poor lighting, and connectivity issues may arise, they can be managed through proper system calibration and maintenance.
Overall, the Smart Box successfully streamlines parcel collection, minimizes manual handling, and enhances security, making it a valuable addition to modern parcel management systems.

Future Scope

The Smart Box for Parcel Collection using RFID Reader and QR Code has the potential for significant advancements in the future. Integrating IoT-based remote monitoring and cloud storage can enhance its functionality, allowing users to track their parcels in real time through mobile applications. Advanced security measures such as facial recognition or biometric authentication can further improve access control, ensuring an even higher level of security. Additionally, incorporating AI-driven analytics can help optimize delivery schedules and predict parcel collection patterns.

The system can be expanded for large-scale deployment in residential complexes, corporate offices, and public parcel lockers. By integrating multiple authentication methods, including OTP-based verification, the smart box can cater to a wider range of users and delivery services. Solar-powered or energy-efficient designs can also be implemented to improve sustainability, making the system more reliable in areas with limited power supply.

REFRENCES

[1]. Pendse, M., Bisht, A., Limaye, S., & Ubale, A. (2022). Baby Monitoring System. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2(1).

[2]. Janney, B., Krishnakumar, S., Anushree, P. B., Rayshma, V., & Suresh, S. (2022). Design of Mobile Infant Incubator with Comforting Pillow. International Journal of Engineering and Technology (UAE), 7(2), 6-9.

[3]. Salahuddin, N. S., Hermita, M., Kurniawan, A., & Basir, D. A. (2022, November 16). Portable Baby Incubator System. Prosiding Seminar Nasional TEKNOIN 2013, 1, A-1. ISBN: 978-602-14272-0-0.

[4]. Salahuddin, N. S., Aryanti, F., Sari, S. P., & Soerowirjo, B. (2021, October 29-30). Wireless Premature Baby Monitoring System. Proceeding 2nd Applied Business and Engineering Conference (ABEC). ISSN: 2339-2053. Politeknik Caltek Riau.

[5]. Fahmi, F., Shalannanda, W., Zakia, I., & Sutanto, E. (2020). Design of an IoT-based smart incubator that listens to the baby. IOP Conference Series: Materials Science and Engineering, 1003, 012153. https://doi.org/10.1088/1757-899X/1003/1/012153

[6]. Kiki, A., Faisal, H., & Junas, H. (2020). Development of a Real-Time Premature Baby Incubator Monitoring System Using Android. Jurnal Amplifier, 10(2). P-ISSN: 2089-2020, E-ISSN: 2622-2000.

[7]. Nugroho, D. P., Munadi, R., & Santoso, I. H. (2021). Internet of Things Based Heart Rate Monitoring System Using ECG Sensor with Android Application Media. e-Proceeding of Engineering, 8(5), 5530–5536.

[8]. Palupi, F. K., Luthfiyah, S., Wisana, I. D. G. H., & Thaseen, M. (2021). Baby Incubator Monitoring Center for Temperature and Humidity using WiFi Network. Journal of Electronics, Electromedical Engineering, and Medical Informatics (JEEEMI), 3(1), 8-

APPENDEX

Appendix-1: Gather Components

1. Collect essential hardware: Microcontroller, RFID Reader, QR Code Scanner, Servo/Solenoid Lock, LCD, Wi-Fi Module.
2. Use RFID tags for parcel identification and generate QR codes for secure recipient access.
3. Ensure availability of power supply, cloud platform access, and monitoring device (PC/Smartphone).

Appendix-2: Circuit Design & Wiring

1. Connect RFID reader and QR scanner to microcontroller (SPI/UART), and lock control to GPIO/Servo pins.
2. Use I2C for LCD display, and connect relay/buzzer to indicate system feedback.
3. Distribute power using a regulated power module and stabilize with capacitors.

Appendix-3: Setting Up the Microcontroller Environment

1. Install Arduino IDE or Micro Python; configure board and serial ports.
2. Add required libraries (RFID, LCD, Wi-Fi, Servo) and enable I2C/SPI interfaces.
3. Test each module (RFID, QR scanner, LCD) individually to confirm setup.

Appendix-4: Writing the Firmware

1. Program RFID reading and QR code authentication logic.
2. Control compartment lock based on valid scans and update LCD display.
3. Send parcel status and logs to cloud platform via Wi-Fi.

Appendix-5: Testing the System

1. Validate RFID detection and QR code matching for proper access control.
2. Check servo/solenoid operation for each locker compartment.
3. Monitor real-time cloud updates and LCD output during operation.

Appendix-6: Install the System in Enclosure

1. Mount electronics in a secured control box with clear display and scanner access.
2. Route wires safely and label locker compartments for ease of use.
3. Provide ventilation and protection for electronic components.

Appendix-7: Final Testing and Optimization

1. Simulate end-to-end usage (delivery and pickup) to check stability.
2. Tune servo angles, scan sensitivity, and power usage for efficiency.
3. Review system logs and cloud dashboard for delays or failures.

Appendix-8: Monitor and Maintain

1. Regularly clean scanner, display, and test locker mechanisms.
2. Check logs for unauthorized access or system errors.
3. Update firmware and replace any worn-out components as needed.

CODE

```
#include <LiquidCrystal.h>
const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#include <SoftwareSerial.h>
SoftwareSerial rfid(A4, A5); // RX, TX
int m1=7;
int m2=6;
void setup()
{
  pinMode(m1,OUTPUT);pinMode(m2,OUTPUT);
  digitalWrite(m1,LOW);digitalWrite(m2,LOW);
  Serial.begin(9600);
  rfid.begin(9600);
  lcd.begin(16, 2);
 lcd.clear(); lcd.print("COURIER DELIVERY");
 lcd.setCursor(0,1);lcd.print("BOX SYSTEM");delay(1000);
}

void loop()
{
  back:
 while(Serial.available())
 {
 String siva=Serial.readString();//Payload:
 int ourdata = siva.indexOf("Payload:");

lcd.clear();lcd.print("QR DATA:");

lcd.print(siva[ourdata+9]);delay(1000);
lcd.print(siva[ourdata+10]);delay(1000);
lcd.print(siva[ourdata+11]);delay(1000);
lcd.print(siva[ourdata+12]);delay(1000);
lcd.print(siva[ourdata+13]);delay(1000);
```

```
if(siva[ourdata+9]=='1'      &&   siva[ourdata+10]=='2'   &&   siva[ourdata+11]=='3'   &&
siva[ourdata+12]=='4' && siva[ourdata+13]=='5')
{
lcd.clear();lcd.print("VALID QRCODE");delay(1000);
digitalWrite(m1,HIGH);digitalWrite(m2,LOW);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
   lcd.clear();lcd.print("BOX OPEN");delay(1000);
   lcd.setCursor(0,1);lcd.print("PIC YOUR PARCEL");delay(1000);

digitalWrite(m1,LOW);digitalWrite(m2,HIGH);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
   lcd.clear();lcd.print("BOX CLOSE");delay(1000);
   lcd.clear();lcd.print("Thank You....");delay(2000);
   lcd.clear();lcd.print("Ready to use....");delay(1000);goto back;
}

 }
  while(rfid.available())
  {
  String rfiddata=rfid.readString();
  Serial.println(rfiddata);
  lcd.clear();lcd.print(rfiddata);delay(1000);
  lcd.setCursor(0,1);lcd.print("Processing.....");delay(1000);
  if(rfiddata=="550014C9E961")
  {
   lcd.clear();lcd.print("COURIER:1");delay(1000);

digitalWrite(m1,HIGH);digitalWrite(m2,LOW);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
   lcd.clear();lcd.print("BOX OPEN");delay(1000);
   lcd.setCursor(0,1);lcd.print("PLACE YOUR PARCEL");delay(1000);

digitalWrite(m1,LOW);digitalWrite(m2,HIGH);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
```

```
    lcd.clear();lcd.print("BOX CLOSE");delay(1000);
    lcd.clear();lcd.print("Thank You....");delay(2000);
    lcd.clear();lcd.print("Ready to use....");delay(1000);goto back;
  }
  else if(rfiddata=="5500147FB48A")
  {
    lcd.clear();lcd.print("COURIER:2");delay(1000);

digitalWrite(m1,HIGH);digitalWrite(m2,LOW);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
    lcd.clear();lcd.print("BOX OPEN");delay(1000);
    lcd.setCursor(0,1);lcd.print("PLACE YOUR PARCEL");delay(1000);

digitalWrite(m1,LOW);digitalWrite(m2,HIGH);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
    lcd.clear();lcd.print("BOX CLOSE");delay(1000);
    lcd.clear();lcd.print("Thank You....");delay(2000);
    lcd.clear();lcd.print("Ready to use....");delay(1000);goto back;
  }
  else if(rfiddata=="5500135D0318")
  {
    lcd.clear();lcd.print("COURIER:3");delay(1000);

digitalWrite(m1,HIGH);digitalWrite(m2,LOW);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
    lcd.clear();lcd.print("BOX OPEN");delay(1000);
    lcd.setCursor(0,1);lcd.print("PLACE YOUR PARCEL");delay(1000);

digitalWrite(m1,LOW);digitalWrite(m2,HIGH);delay(3000);digitalWrite(m1,LOW);digita
lWrite(m2,LOW);
    lcd.clear();lcd.print("BOX CLOSE");delay(1000);
    lcd.clear();lcd.print("Thank You....");delay(2000);
    lcd.clear();lcd.print("Ready to use....");delay(1000);goto back;
  }
```

```
  else
  {
   lcd.clear();lcd.print("INVALID CARD");delay(1000);
  }


  }
}
```