A

MAJOR PROJECT REPORT

ON

ACCESS TO THE UNKNOWN VEHICLE INTO THE PARTMENT THROUGH THE QR SCANNER AND OTP.

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

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

MD. ZAKIR	**218R1A04P6**
SK. ABDUL RAVUF	**228R5A0422**
S. KAVYA	**228R5A0423**
S. ARAVIND PRASAD	**228R5A0424**

Under the Esteemed Guidance of

DR. K. SRAVAN ABHILASH

ASSOCIATE PROFESSOR, ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)

Kandlakoya(V), Medchal(M), Telangana – 501401

(2024-2025)

CMR ENGINEERING COLLEGE

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accreditedby NBA)

Kandlakoya (V), Medchal Road, Hyderabad - 501 401

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that Major project work entitled **"ACCESS TO THE UNKNOWN VEHICLE INTO THE APARTMENTS THROUGH THE QR SCANNER AND OTP"** is being Submitted by **MD. ZAKIR** bearing Roll No: **218R1A04P6, SK. ABDUL RAVUF** bearing Roll No**: 228R5A0422, S. KAVYA** bearing Roll No: **228R5A0423, S. ARAVIND PRASAD** bearing Roll No: **228R5A0424** in B.Tech IV-II semester, Electronics and Communication Engineering is a record bonafide work carried out by then during the academic year 2024-25.

INTERNAL GUIDE:	**HEAD OF THE DEPARTMENT**
DR. K. SRAVAN ABHILASH	**Dr. SUMAN MISHRA**
Associate Professor, ECE	

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

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 major 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, throughout the course of this work. We sincerely thank our project internal guide **DR. K. SRAVAN ABHILASH,** Associate Professor of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We hereby declare that the Major project entitled **“ACCESS TO THE UNKNOWN VEHICLE INTO THE APARTMENTS THROUGH THE QR SCANNER AND OTP”** 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.**

MD. ZAKIR (218R1A04P6)

SK. ABDUL RAVUF (228R5A0422)

S. KAVYA (228R5A0423)

S. ARAVIND PRASAD (228R5A0424)

ABSTRACT

In today's era of smart living, secure access to vehicles and residential spaces is paramount. This paper presents an innovative system for generating and managing automatic passwords for accessing unknown vehicles and apartments. The proposed system leverages advanced encryption techniques and machine learning algorithms to ensure robust security and user convenience. The core component of the system is an automatic password generator that creates unique, time-sensitive passwords for each access attempt. These passwords are generated based on a combination of user authentication, environmental factors, and predefined security protocols. The system integrates with existing smart locks and vehicle security systems, providing seamless and secure access.

The influx of unknown vehicles into apartment premises poses significant security challenges. This abstract proposes a robust access control system that utilizes QR code scanning and One- Time Password (OTP) verification to manage and monitor the entry of unknown vehicles. A graphical user interface can be created on the IOT web server side. The graph visualizes the sensor data in a convenient way. By virtue of this IOT based pollution monitoring system project, air and pollution levels can be constantly monitored from a remote location. This remote location can be anywhere in the world. We can then take steps in order to reduce pollution levels.

The advent of technology has opened new avenues for enhancing security measures in residential areas. Traditional methods of manual vehicle verification are often inefficient and prone to human error. To address these issues, the integration of QR scanners and OTP systems offers a high-tech solution that ensures both efficiency and security.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

In an era marked by rapid technological evolution and escalating security threats, the imperative to innovate protective measures within residential environments has become paramount. The advent of digital technologies has presented new opportunities to bolster the security infrastructure of urban living spaces, notably through the integration of intelligent systems that can adeptly manage access controls with unprecedented precision and adaptability. Among these innovations, the application of automatic number plate recognition technology stands out as a revolutionary method to enhance both the security and convenience of apartment complexes.

The crux of this advanced system lies in its capacity to seamlessly blend cutting-edge technology with the practical demands of apartment security management. The system employs automatic number plate recognition technology, a mechanism that scrutinizes the number plates of vehicles as they attempt entry into the complex. This technology is not merely reactive but proactive in its capability to compare these captured number plates against a comprehensive database of vehicles that are authorized to access the premises. Vehicles that are successfully matched are allowed entry, which exemplifies the system's role in reinforcing security without sacrificing the ease of entry for authorized residents.

1.2 OBJECTIVE OF THE PROJECT

The objective of this project is to design and implement a secure, efficient, and user-friendly access control system for managing the entry of unknown vehicles into apartment complexes. By leveraging QR code scanning and One-Time Password (OTP) verification, the project aims to enhance the overall security infrastructure, streamline vehicle access procedures, and provide comprehensive tracking and monitoring capabilities This project is to develop an advanced IoT-based air pollutants monitoring system that facilitates real-time detection and analysis of various air pollutants to support health and safety. By using a comprehensive range of sensors—covering gases like ozone, carbon monoxide, methane, LPG, and other volatile organic compounds (VOCs), as well as particulate matter (PM2.5 and PM10)—the system captures a holistic view of environmental air quality. The integration of a DHT22 sensor for temperature and humidity measurement enhances the data accuracy by providing context for air pollutant readings.

The primary objective of this project is to establish a secure, efficient, and user-friendly system to manage and monitor the entry of unknown vehicles into apartment complexes. Leveraging the dual technologies of QR code scanning and One-Time Password (OTP) verification, the project aims to significantly bolster security by ensuring that only vehicles with authorized credentials can gain entry. This robust security measure reduces the risk of unauthorized access and enhances the overall safety of residents. Moreover, the system is designed to streamline the entry process, reducing the need for manual intervention, thereby minimizing wait times and easing congestion at entrance gates. This efficient automation improves the flow of vehicles, enhancing the daily experience for both residents and visitors.

Additionally, the system will provide comprehensive tracking and monitoring capabilities by maintaining detailed logs of all vehicle entries and exits. This feature allows security personnel to conduct thorough audits and real-time monitoring, ensuring a high level of oversight. The system also prioritizes user convenience by offering an intuitive interface for generating and managing QR codes and OTPs, making the process simple and accessible for residents and visitors alike. Furthermore, the system's adaptable and scalable design ensures it can be implemented across various apartment complexes, accommodating different levels of security requirements and user demands. Ultimately, this project aims to create a robust, reliable, and user-centric access control mechanism that enhances security, improves operational efficiency, and ensures a safe and convenient environment for apartment complex residents.

The project's key features include QR code scanning, OTP generation, and access control systems. Our team leveraged these cutting-edge technologies to develop a secure, convenient, and innovative access control system.The successful completion of this project demonstrates our team's capabilities in designing and implementing innovative solutions to real-world problems. At the core of this system is the utilization of automatic number plate recognition (ANPR) technology. This technology serves as the primary mechanism for monitoring and controlling vehicle entry. ANPR systems function by capturing the number plates of vehicles attempting to enter the complex and immediately comparing these plates with a pre-established The sophistication of the ANPR technology lies in its high-resolution cameras and advanced image processing algorithms, which can accurately read number plates under various lighting and weather conditions.

1.3 ORGANIZATION OF THE PROJECT

The project's core objective was to design and implement a system that grants access to authorized vehicles using QR code scanning and one-time password (OTP) verification. This ensures that only approved vehicles can enter the apartment complex, thereby minimizing the risk of unauthorized entry. Our team leveraged cutting-edge technologies, including QR code scanning, OTP generation, and access control systems, to develop this innovative solution. Through this project, we demonstrated our expertise in designing and implementing secure and convenient access control systems.

To achieve this objective, our team developed a system that utilizes QR code scanning and one-time password (OTP) verification to grant access to authorized vehicles. This innovative solution ensures that only approved vehicles can enter the apartment complex, thereby enhancing security and minimizing the risk of unauthorized entry.The project's key features include QR code scanning, OTP generation, and access control systems. Our team leveraged these cutting-edge technologies to develop a secure, convenient, and innovative access control system.The successful completion of this project demonstrates our team's capabilities in designing and implementing innovative solutions to real-world problems.

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 and is still largely in use today.

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.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

System Overview for Unknown Vehicle Access

1. Entry Request Process:

Visitor Registration:

Unknown vehicle drivers are required to register their details (name, phone number, purpose of visit, vehicle number) with security personnel or via a resident app/visitor kiosk at the gate.

Resident Authorization:

The resident being visited is notified and must approve the visitor. Approval can be done via a mobile app or SMS.

2. QR Code/OTP Generation:

On Approval:

Upon the resident's confirmation, the system generates a unique QR code or OTP tied to the visitor's vehicle and sends it via SMS or email to the driver.

Temporary Validity:

QR codes/OTPs are time-bound and can only be used for the authorized period.

3. Gate Access Using QR or OTP:

QR Code Scanning:

The vehicle driver scans the QR code at a gate-mounted scanner. The system validates the code and opens the gate if the data matches the database.

OTP Entry:

Alternatively, the driver enters the OTP on a keypad or security device at the gate. A successful match grants entry.

4. Vehicle Entry Logging:

The system logs the vehicle's details, entry time, and driver information for monitoring and security purposes.

Residents and security staff can access these logs in real time

Implementation in Existing System

2.2 PROPOSED SYSTEMS

In the contemporary landscape of urban living, where the dynamics of security and convenience continually evolve, the need for innovative solutions to safeguard residential areas has never been more pressing. Amidst this backdrop, our project introduces a groundbreaking system designed to revolutionize the way apartment complexes manage vehicle access, setting new benchmarks for security measures integrated with advanced technological capabilities.The proposed system, "Access to the unknown vehicle into the apartment through the automatic password generator," is meticulously engineered to enhance the security and operational efficacy of apartment complexes.

At the core of this system is the utilization of automatic number plate recognition (ANPR) technology. This technology serves as the primary mechanism for monitoring and controlling vehicle entry. ANPR systems function by capturing the number plates of vehicles attempting to enter the complex and immediately comparing these plates with a pre-established The sophistication of the ANPR technology lies in its high-resolution cameras and advanced image processing algorithms, which can accurately read number plates under various lighting and weather conditions. This capability is vital for maintaining uninterrupted security operations around the clock. Once a vehicle's number plate is captured, the system cross-references it with the database. If a match is found, the barriers are automatically lifted, allowing the vehicle to enter without any human intervention.

This seamless operation not only enhances the security by minimizing human error but also significantly speeds up the entry process, thus enhancing convenience for the residents. However, a standout feature of our system is its approach to handling situations where a vehicle's number plate does not match any in the database. In such cases, rather than outright denying access, which could inconvenience legitimate visitors or new residents, our system employs an innovative automatic password generator. This feature is designed to provide

ANPR systems function by capturing the number plates of vehicles attempting to enter the complex and immediately comparing these plates with a pre-established The sophistication of the ANPR technology lies in its high-resolution cameras and advanced image processing algorithms

2.3 EMBEDDED 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,

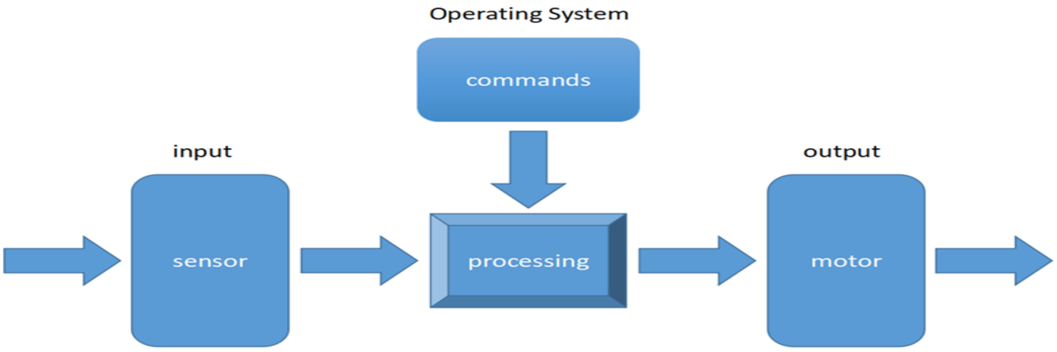

FIG: 2.1 EMBEDDED OS

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. User interface scan include buttons, LEDs and touchscreen sensing. Some systems use remote user interfaces as well.

History of embedded systems

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.

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 and is still largely in use today.

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.

Characteristics of embedded systems

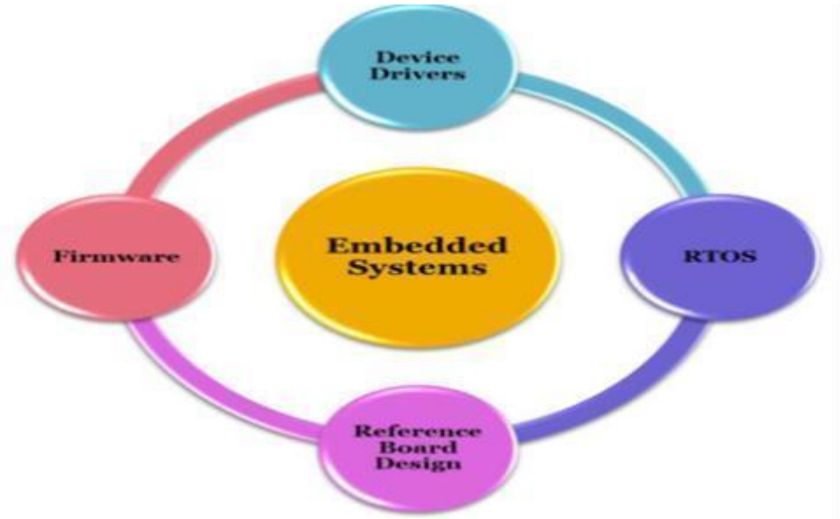

FIG: 2.2 EMBEDDED SYSTEMS

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 andmicrocontrollers. 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.

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.
- 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: The hardware of embedded systems is based around microprocessors and micro controllers. 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: 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 the software that interfaces directly with the hardware.

FIG: 2.3 BLOCKS OF EMBEDDED SYSTEMS

2.3.1 WHY EMBEDDED?

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.

Embedded Systems has brought about a revolution in Science. It is also a part of a 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.

Well this is just one good thing about IoT. We can monitor Pollution Levels, we can control the intensity of street lights 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 SYSTEMS HARDWARE

2.3.2 DESIGN APPROACHES

A system designed with the embedding of hardware and software together for a specific function with a larger area is an 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 and less power consumption. The application of microcontrollers is MP3 and 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 theamount of software accommodated within CES has considerably changed. For example in smart cars the amount of software has grown about 100 times compared to previousyears. This change means that software design for these systems is also bounded to hardconstraints (e.g., high security and performance). Along the evolution of CES, the approaches for designing them are also changing rapidly, so as to fit the specialized needs of CES. Thus, a broad understanding of such approaches is missing.

Steps in the Embedded System Design Process

The different steps in the embedded system design flow/flow diagram include the following.

- **Specification**: The first step in the process, where you define the requirements that the system must meet

Hardware and software partitioning: You divide the system into hardware and software components

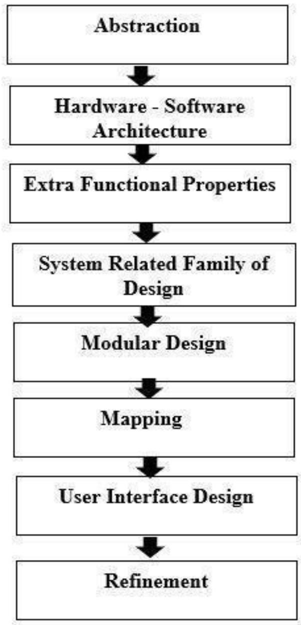

FIG: 2.5 EMBEDDED DESIGN-PROCESS-STEPS

- **Hardware and software design**: You design approach the hardware and software independently
- **Hardware and software integration**: You integrate the hardware and software, and decide how and when to resolve bugs
- **Software testing**: You test the software to detect vulnerabilities
- **User interface design**: You design the interface between the CPU software and the digital interface logic, and between the digital and analog sides of the interface

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.

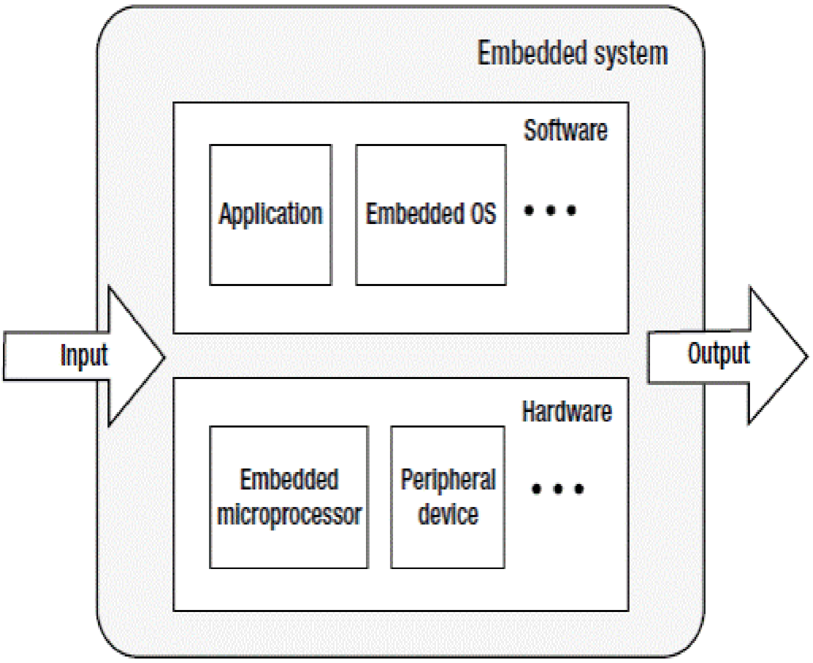

FIG: 2.6 HARDWARE AND SOFTWARE OF EMBEDDED SYSTEM

Table: 2.1 DESIGN PARAMETERS AND FUNCTIONS OF AN EMBEDDED SYSTEM

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.

Architectural description language is used to describe the software design.

• Control Hierarchy

• 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.WHO has identified formulations for their local preparation. Embedded systems are used in a variety of technologies across industries. Some examples include:

Automobiles: Modern cars commonly consist of many computers 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.

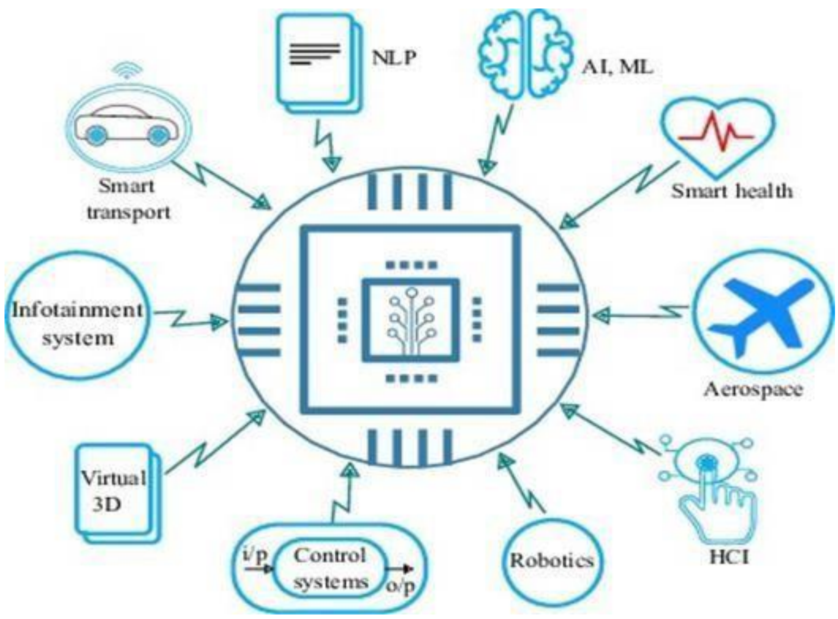

FIG: 2.7 APPLICATIONS OF EMBEDDED SYSTEMS

- **Mobile phones.** These consist of many embedded systems, including GUI software and hardware, operating systems, cameras, microphones and USB I/O modules.
- **Industrial machines.** They 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 userfriendly, 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 handrub 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.

2.3.3 COMBINATION OF LOGIC DEVICES

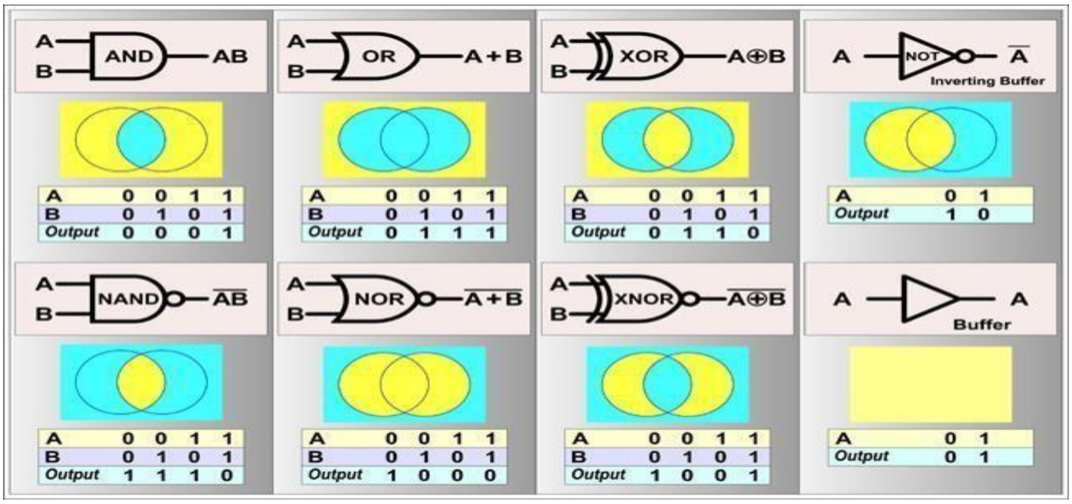

FIG: 2.8 LOGIC GATES

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.

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.

2.4 ARDUINO

Overview The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

- Stronger RESET circuit.
- At mega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

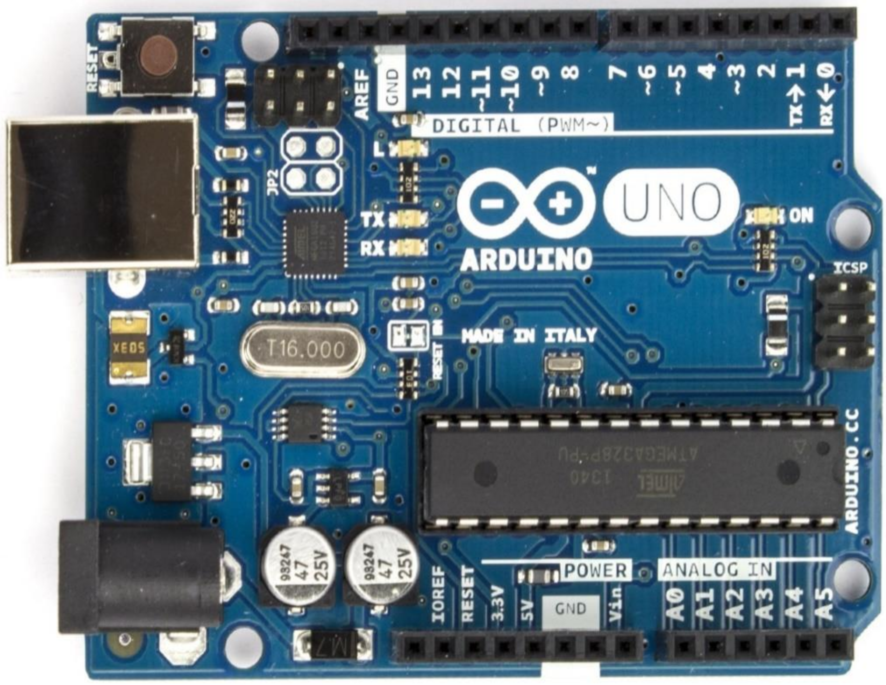

Fig 2.9 Arduino

SUMMARY:

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

2.5 POWER SUPPLY

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.

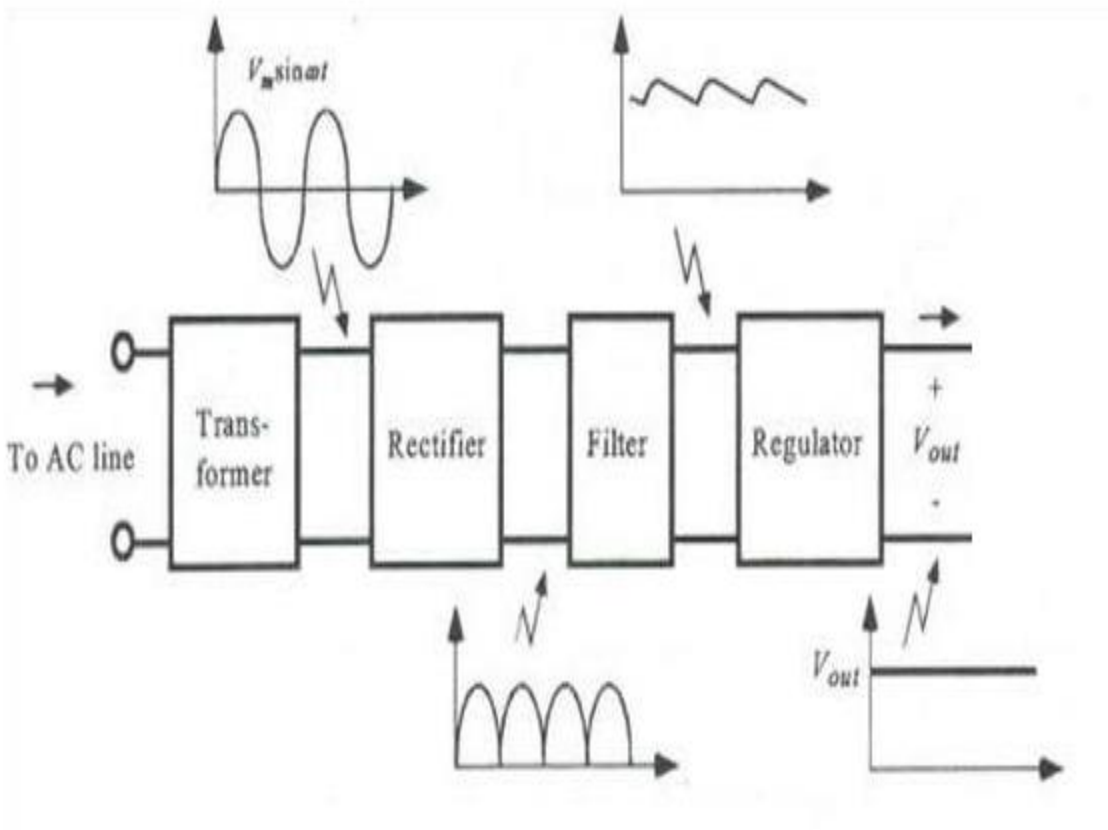

Fig 2.10 Power Supply

Designing a power supply for an embedded kit is a critical aspect of ensuring the reliability and efficiency of the system. The first consideration is understanding the voltage requirements of the components involved, such as microcontrollers, sensors, and communication modules.

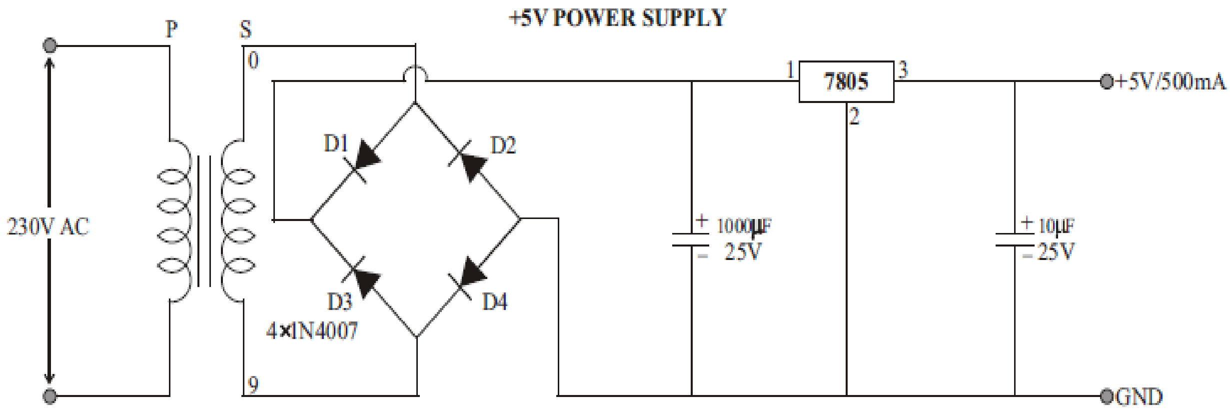

Fig 2.11 Circuit Diagram of Power supply

Some may require 3.3V, while others need 5V or more, necessitating a power supply capable of meeting these varied needs. It must also be able to supply adequate current to handle the combined demands of all components, ensuring no interruptions in functionality.

Efficiency is particularly important, especially for battery-powered applications. To extend battery life, switching regulators like buck converters are often used, as they are more energy-efficient than linear regulators. Noise filtering is another crucial factor in the design process. Power supply noise can interfere with sensitive components, potentially causing erratic behavior or inaccuracies. Careful selection of capacitors and inductors for decoupling and filtering can help mitigate this issue.

Finally, the choice of power source plays a significant role. Depending on the application, the power supply may need to support wall power, battery operation, or a combination of both. Battery-powered systems might require additional features like low-dropout regulators or battery management circuits. With proper planning and design, a well-built power supply ensures that the embedded kit operates seamlessly and meets the demands of its intended application. Let me know if you'd like to delve deeper into any of these aspects!

STEP DOWN TRANSFORMER:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. There fore a regulator is applied at the output stage.

Voltage Regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

Features:

- Output Current up to 1A.
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V.
- Thermal Overload Protection.
- Short Circuit Protection.
- Output Transistor Safe Operating Area Protection.

CHAPTER 3
SOFTWARE REQUIREMENTS

3.1 SOFTWARE TOOLS

Embedded system software

A typical industrial microcontroller is unsophisticated compared to the typical enterprise desktop computer and generally depends on a simpler, less-memory-intensive program environment. The simplest devices run onbare metaland are programmed directly using the chip CPU's machine code language.

Often, embedded systems use operating systems or language platforms tailored to embedded use, particularly where real-time operating environments must be served. At higher levels of chip capability, such as those found in SoCs, designers have increasingly decided the systems are generally fast enough and the tasks tolerant of slight variations in reaction time that near- real-time approaches are suitable. In these instances, stripped-down versions of the Linux operating system are commonly deployed, although other operating systems have been pared down to run on embedded systems, includingEmbedded Javaand Windows IoT (formerly Windows Embedded).

Generally, storage of programs and operating systems on embedded devices make use of either flash or rewritable flash memory.

These activities include:

- Setting up a cloud service provider such as Amazon Web Services, Google Cloud, etc
- Set up private and public keys along with a device certificate.
- Write a device policy for devices connecting to the cloud service
- Connect an embedded system to the cloud service
- Transmit and receive information to the cloud
- Build a basic dashboard to examine data in the cloud and control the device

If developers are able to do these things, they will have built a good foundation from which to master cloud connectivity for their embedded systems.

Software requirements

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 run time libraries. Operating system is one of the requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of 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-enddisplay adapters, needs specialAPIor newer device drivers. A good example isDirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, onMicrosoftplatforms.

Web browser

Mostweb applicationsand software depend heavily onweb technologiesto make use of the default browser installed on the system.Microsoft Internet Exploreris a frequent choice of software running onMicrosoft Windows, which makes use ofActiveXcontrols, despite their

- Raspberry is an open-source software that is mainly used for writing and compiling the code into the raspberry.
- It is an official raspberry, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

- It is easily available for operating systems like MAC, Windows, Linux and runs on the Java, python Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

- The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT).

- Each of them contains amicrocontrolleron the board that is actually programmed and accepts the information in the form of code.

The Raspberry Pi is a low-cost, credit-card-sized single-board computer that has become immensely popular for education, DIY projects, prototyping, and hobbyist electronics. Developed by the Raspberry Pi Foundation.

Programming Languages: The Raspberry Pi supports a variety of programming languages, including Python, C, C++, Java, and many others. Python is particularly popular for educational purposes.

3.2 RESEARCH

The embedded systems industry was born with the invention of microcontrollers and since then it has evolved into various forms, from primarily being designed for machine control applications to various other new verticals with the convergence of communications. Today it spans right form small metering devices to the multi-functional smartphones. I will cover the areas that are currently focused for development in embedded systems and state what are the ongoing research opportunities in that particular area.

Security

Security remains a great challenge even today. Ongoing Research is to sustain physical tampering, mechanisms to trust the software, authenticate the data and securely communicate over internet. With the advent of IoT/IoE, not only the number devices will continue to increase but also will the **number of possible attack vectors**. Many challenges remain ahead to get the connected devices on a billion scale.

Connectivity

Wi-Fi, BLE, ZigBee, Thread, ANT, etc have been adapted by embedded system experts from considerable time. Head-on competition between these groups is in progress to determine as to who will emerge as the best solution provider to this huge estimated market of IoT/IoE.

4G/5G on low power devices is the ongoing experimentation which will make embedded systems easily and robustly connect to the internet. Communication using GSM/LTE in licensed/unlicensed communication bands with the cloud can change the ball game of IoE all together.

Memory

Various type of volatile/non-volatile memories with variable sizes and speeds are widely available today. Research is more towards **organizing** them in best possible architecture to reach closer to the design goal of optimal power-performance-cost

Energy

Power/Battery management has been under focus for some time. Usage of **renewable resources** to power device's lifetime is currently the challenge that is tried to address; especially for wearables. Optimal power usage to get **Longer Battery Life** with new Hardware/Software architectural designs will continue for some time.

System

Multicore (Symmetric/Asymmetric) architectures are experimented since long. Addition of **GPUs** to systems for VR/Gaming/Machine learning is addressed currently.

Programmable SOCs (**PSOCs)** - (Configurable Hardware Capability) have been there for a long time now, but some has not yet gained momentum. Application-specific computer architectures is also in the pipeline in order to optimize the design matrix of power-performance-cost.

Performance

Real-time on-board Image/Video/Audio processing, feature enabled cameras, on board machinelearning are all currently experimented with varied approaches. Commercialization of these technologies has already started but there is still some time to get the best out of these technologies and there is lot of scope to make them more userfriendly

Other than this, hardening of modular software functionalities (Yes lot of architectures are coming up with hardware performing redundant software functionalities). Ongoing research is to analyse the performance and determine the applications where this strategy can be fruitful.

Networking

Wireless Sensor Networks, Machine to Machine Communication/Interaction, Human

Computer Interaction, Security Gateway protocols are still being improved. Light weight algorithms with optimal security will be targeted for embedded systems.

Real Time Operating Systems (RTOS)

Many companies are backing at least one Real Time Open-Source Operating System and there are many out there. Challenge is to cover the wide span of devices, there functionalities and variety of applications.

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. 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.

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

Real-Time Operating Systems (RTOS) are specialized operating systems designed to handle real-time tasks with precision and reliability. They are widely used in embedded systems where meeting specific timing requirements is critical, such as in automotive systems, medical devices, robotics, and aerospace applications. Unlike general-purpose operating systems, an RTOS ensures that tasks are executed within defined time constraints, providing predictable and deterministic behavior. This feature makes RTOS essential for safety-critical systems where delays or unpredictability can lead to significant consequences.

CHAPTER 4

HARDWARE REQUIREMENTS

4.1 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 realtime computing.

Microprocessors are visually indistinguishable from microcontrollers. However, the microprocessor only implements a central processing unit (CPU) and, thus, 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.

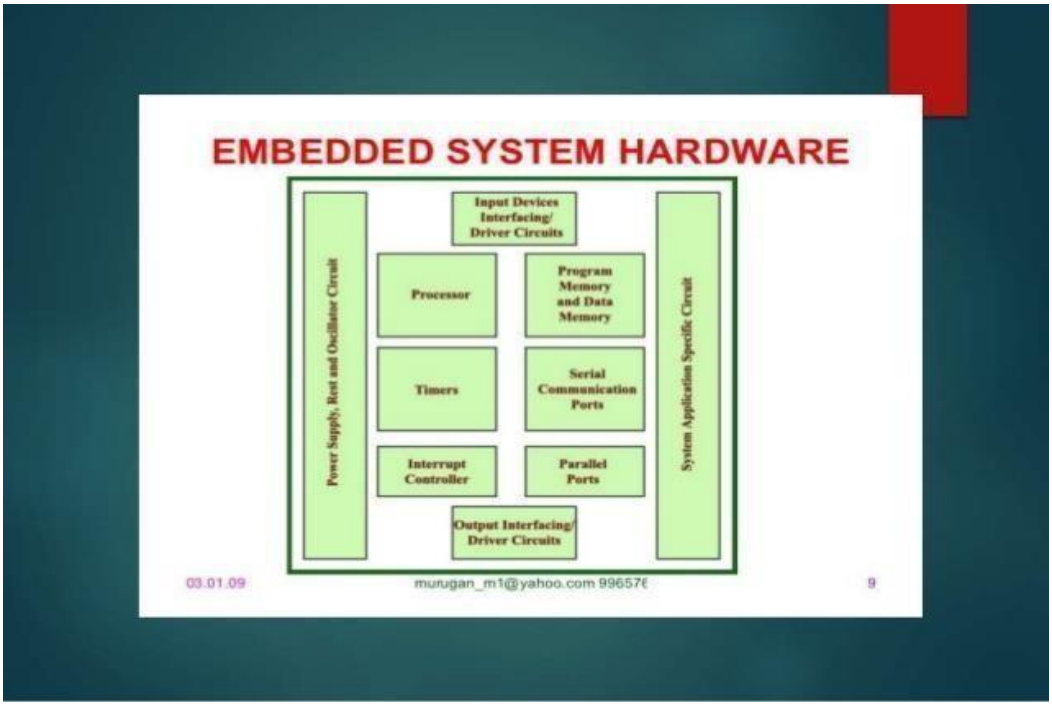

FIG: 4.1 EMBEDDED SYSTEMS HARDWARE BLOCK DIAGRAM

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. 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. 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.

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 general of this first definition, giving the requirements to be met in the design of a system or subsystem.

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.

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.

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 onx86 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-endgames, 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 −

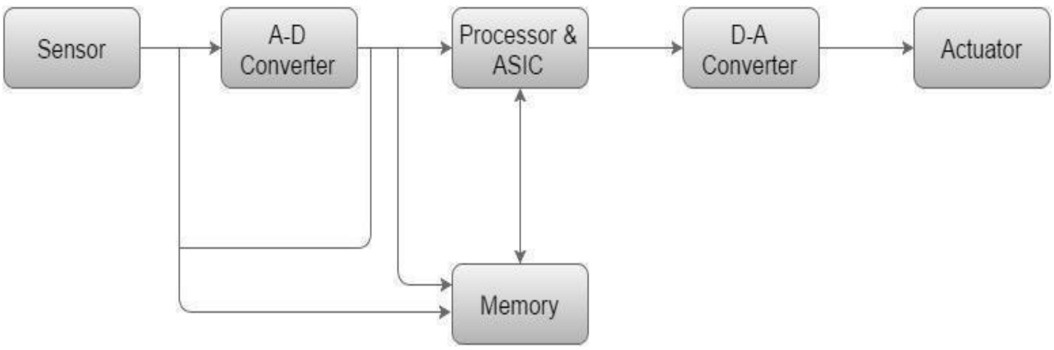

FIG: 4.2 PERIPHERALS OF EMBEDDED SYSTEMS

4.2 BLOCKDIAGRAM

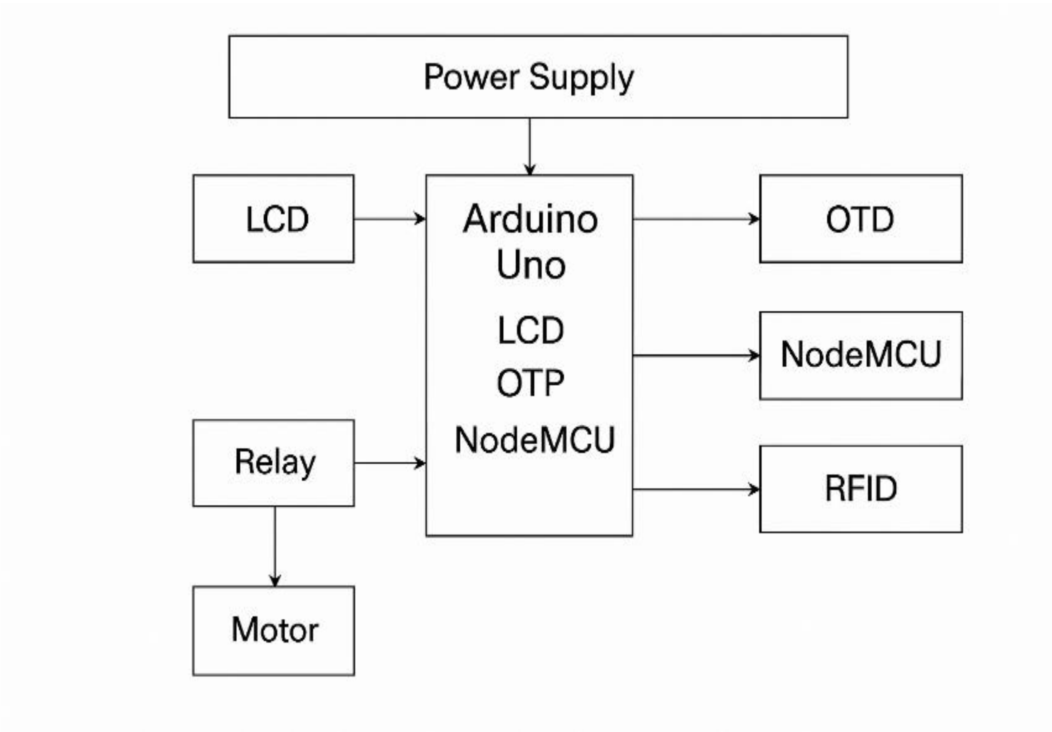

FIG: 4.3 BLOCK DIAGRAM

4.3 WORKING

The system functions in the following sequence:

QR Code Scanning:

- When an unknown vehicle arrives at the apartment gate, the visitor scans a QR code displayed at the entrance using their mobile device.
- The QR code triggers the system to initiate the authentication process.

OTP Generation and Verification:

- Upon scanning the QR code, the system sends a request to the server via Node MCU to generate a One-Time Password (OTP).
- The OTP is sent to the authorized resident's registered mobile device.
- The visitor is prompted to enter the OTP on the system interface.
- Arduino UNO verifies the entered OTP with the server. If the OTP matches, the visitor is authenticated.

RFID Authentication (For Registered Vehicles):

- For registered residents, RFID tags are used for seamless access.
- When an RFID tag is detected, the Arduino UNO checks the tag's authenticity.
- If authenticated, the gate opens automatically without requiring OTP verification.

Gate Control Mechanism:

- If the OTP is verified or RFID authentication is successful, Arduino UNO activates the motor through a relay system to open the gate.
- After a predefined time, the gate automatically closes to maintain security.

Real-Time Monitoring and Alerts:

- Node MCU is connected to the internet for real-time monitoring of vehicle entries and exits.
- The system logs all vehicle access events and sends alerts or notifications to the security personnel and residents, ensuring transparent access control.

User Interface and Feedback:

- The LCD display provides real-time feedback to the visitor, such as "Enter OTP," "Access Granted," or "Access Denied."
- This enhances the user experience and guides the visitor through the authentication process.

4.4 Component Integration and Communication:

QR Scanner and Arduino UNO:

- The QR scanner communicates with Arduino UNO to initiate the authentication process.
- Upon successful scanning, Arduino sends a request to Node MCU for OTP generation.

Node MCU and OTP Module:

- Node MCU is responsible for sending the OTP request to the server and receiving the OTP.
- The OTP is then sent to the authorized resident's mobile device for verification.

RFID Module and Arduino UNO:

- The RFID module continuously scans for registered tags.

- Upon detecting a registered tag, Arduino verifies its authenticity and grants access.

Motor and Relay System:

- Arduino UNO controls the motor through a relay switch to open and close the gate.
- The motor mechanism is designed to operate smoothly and efficiently, ensuring safe gate movement.

LCD Display and Arduino UNO:

- The LCD display is connected to Arduino UNO to provide real-time system status updates and instructions to the visitor.

Output:

Access granted or denied based on the input validation.

4.5 INTRODUCTION TO SENSORS

Sensors are essential devices that detect and respond to changes in physical, chemical, or biological conditions. They convert these changes into electrical signals that can be measured, analyzed, and used in various applications. Sensors play a critical role in modern technology, enabling automation, data collection, and intelligent decision-making.

4.5.1 MQ Sensors

The "MQ" in MQ sensors stands for "Metal Oxide". The sensors are based on a metal oxide semiconductor (MOS) technology, where the sensing element consists of a metal oxide layer that reacts with specific gases. When these gases come into contact with the sensor, they cause a change in the resistance of the metal oxide layer, which can be measured to determine the concentration of the target gas.

FIG: 4.4 MQ Sensors

The MQ-131 is an ozone (O_3) gas sensor that detects ozone levels by measuring changes in the resistance of its metal oxide semiconductor material. It is commonly used in air quality monitoring systems, especially for detecting ozone in industrial or urban environments. It operates by using a metal oxide semiconductor (MOS) that changes resistance when exposed to ozone, allowing the concentration of the gas to be measured. This sensor is often used in environmental monitoring and safety applications.

The MQ-2 is a versatile sensor that detects a variety of gases including smoke, LPG, methane, butane, and carbon monoxide, with its resistance varying based on exposure to these gases. This sensor is often used in smoke alarms, gas leak detectors, and air quality monitoring. Its resistance changes when exposed to these gases, making it suitable for applications such as smoke alarms, gas leakage detection in homes and industries, and air quality monitoring in urban areas. Its broad sensitivity to different gases makes it very versatile for general gas detection purposes.
The MQ-7 is specifically designed to detect carbon monoxide (CO), utilizing a sensitive film that reacts to CO, causing resistance changes which help determine its concentration. It is widely used in carbon monoxide detection systems, particularly in homes and vehicles. It operates by detecting the chemical changes that occur when the sensor's metal oxide surface interacts with CO molecules, resulting in a measurable change in resistance. The MQ-7 is primarily used in carbon monoxide detectors, especially in residential and automotive applications, where CO is a significant concern. Finally, the MQ-4 is designed for detecting methane (CH_4) gas, with its resistance changing in response to the presence of methane. This sensor is typically used in gas leak detection systems in areas such as natural gas pipelines and industrial plants. The MQ-4 sensor works similarly to other MQ sensors by detecting changes in resistance when exposed to methane. It is widely used in gas leak detection systems, particularly in natural gas pipelines, industrial plants, and areas where methane is used or stored.

The analog output provides a continuous voltage signal proportional to the gas concentration, while the digital output can be used for threshold-based detection. The sensor operates at a voltage of 5V and requires a preheating time of approximately 20 seconds before it can provide accurate readings.One of the key advantages of the MQ-6 sensor is its high sensitivity to LPG and butane, which makes it ideal for detecting gas leaks in residential and commercial settings. Additionally, its low cost and ease of integration into existing systems make it an attractive option for developers and engineers. However, the sensor's performance can be affected by environmental factors such as temperature and humidity, which may require calibration for optimal accuracy.

The MQ-6 sensor is often used in conjunction with alarm systems to provide real-time alerts in case of gas leaks. This integration enhances safety by enabling quick responses to potential hazards. The sensor's applications extend beyond gas leakage detection to include industrial automation, environmental monitoring, and even IoT-based smart home systems.

In summary, the MQ-6 gas sensor is a reliable and efficient tool for detecting combustible gases like LPG and butane. Its combination of sensitivity, affordability, and versatility makes it a valuable component in various safety and monitoring applications. As technology continues to advance, the MQ-6 sensor remains a cornerstone in the development of gas detection systems.

Introduction to PMS5003 Sensor

The PMS5003 is a popular particulate matter sensor used to measure the concentration of particulate matter (PM) in the air. It specifically detects PM2.5 (particles with a diameter of 2.5 micrometers or smaller) and PM10 (particles with a diameter of 10 micrometers or smaller), which are key indicators of air quality.

The sensor utilizes laser scattering technology to detect and measure these fine particles in real time. When air particles pass through the laser beam, they scatter light, and the sensor detects this scattered light to determine the concentration of particulates in the air.

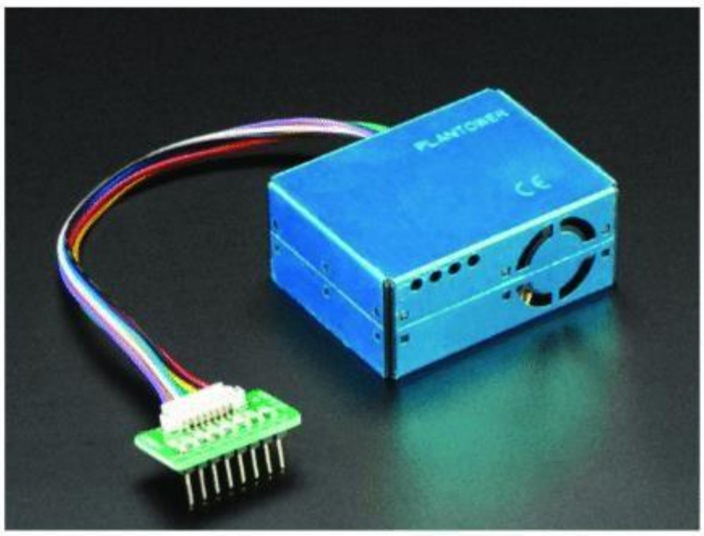

FIG: 4.5 PMS5003 SENSOR

The PMS5003 sensor provides data in terms of micrograms per cubic meter (µg/m³), which is the standard unit for measuring particulate concentration. It offers high accuracy and low power consumption, making it ideal for use in portable or embedded systems, such as air quality monitors, IoT-based devices, and environmental monitoring setups.

One of the key features of the PMS5003 is its ability to provide measurements for multiple particle sizes. It can measure PM1.0, PM2.5, and PM10 concentrations, which are commonly used to evaluate air quality in homes, offices, factories, and outdoor environments. PM2.5, in particular, is a significant health concern as these particles are small enough to enter the lungs and bloodstream, potentially leading to respiratory and cardiovascular issues. Overall, the PMS5003 sensor is a reliable and widely used tool for detecting airborne particulate matter and plays a crucial role in monitoring and improving air quality, particularly in urban and industrial environments.

These pollutants impact respiratory health, and AQI sensors aggregate their readings to provide a standardized index value representing air quality on a scale (typically 0-500), with lower values indicating healthier air and higher values indicating more hazardous conditions. AQI sensors are commonly used in environmental monitoring systems, personal air quality devices, and smart city applications to help individuals and authorities monitor air pollution levels in real-time, take preventive actions, and implement policies for public health and environmental safety.

In more advanced IoT monitoring systems, where multiple sensors and communication modules (like GSM or Wi-Fi modules) need to work simultaneously, it is essential to manage power distribution effectively. Using a regulated power supply ensures that each component receives a consistent and adequate power level to function optimally. Power management circuits such as voltage regulators, DC-DC converters, and battery management systems can be used to optimize power consumption, extend battery life, and prevent overloading or damage to sensitive components, ensuring the long-term performance and stability of the monitoring system. 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. 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.

4.5.1 Introduction to DHT22 Sensor

Voice recognition, also known as speech recognition, is a computer technology that analyzes and converts spoken words or commands into editable text or executable actions. It enables individuals to interact with computers or devices through speech, eliminating the need for traditional input methods like keyboards or mice. Voice recognition technology has found applications in various areas, including voice assistants, smart homes, voice search, and voice-controlled notebooks.

The DHT22(also known as the AM2302) is a popular digital sensor that measures temperature and humidity. It is widely used in DIY projects, environmental monitoring, and home automation due to its affordability and reasonable accuracy. The sensor can measure temperatures ranging from -40°C to 80°C with an accuracy of ±0.5°C, and it can measure humidity levels from 0% to 100% with an accuracy of ±2-5% RH.

It provides a digital output, simplifying integration with microcontrollers like Arduino or Raspberry Pi. The DHT22 operates on a 3.3V to 6V power supply and consumes low power, making it ideal for battery-operated systems. It typically provides readings every 2 seconds, which makes it suitable for real-time monitoring. It is commonly used in weather stations, HVAC systems, agriculture for greenhouse monitoring, and air quality monitoring systems. The DHT22 is appreciated for its ease of use, low cost, and reliability in a wide range of applications where temperature and humidity need to be tracked.

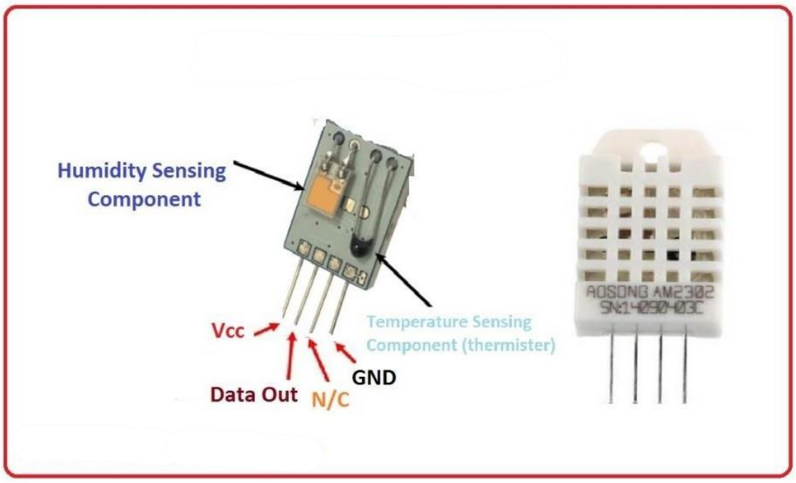

FIG: 4.6 DHT22 Sensor

4.5.2 Introduction to MICS Sensor

The MiCS-2714 sensor is designed to detect Nitrogen Dioxide (NO_2) in the range of 0.05 to 5 ppm, making it suitable for urban air quality monitoring. It operates across a wide temperature range (-30°C to +85°C) and responds quickly to NO_2 due to its metal oxide semiconductor technology.

The MiCS-5524, on the other hand, is a Carbon Monoxide (CO) sensor. It detects CO concentrations in the range of 1 to 1000 ppm, ideal for applications needing CO monitoring, like indoor air quality systems or industrial safety. It shares a similar operating temperature range and utilizes metal oxide technology for reliable and fast detection of CO levels.

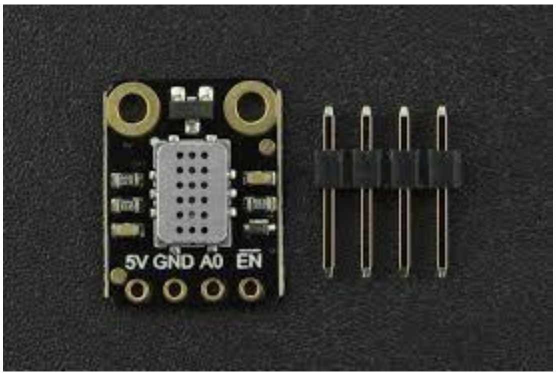

FIG:4.7 MICS-5524

The MiCS-2714 and MiCS-5524 sensors are both compact, highly sensitive metal oxide gas sensors, often integrated into systems for air quality and pollution monitoring. Here's a closer look at each:

MiCS-2714: NO_2 Sensor

The MiCS-2714 detects Nitrogen Dioxide (NO_2), a harmful pollutant commonly emitted from vehicle exhaust, industrial processes, and combustion activities. NO_2 is a significant component of urban smog and can adversely affect respiratory health. Key specifications include:

- Measurement Range: 0.05 to 5 ppm, which is suitable for urban and industrial monitoring.
- Sensitivity and Stability: This sensor is highly sensitive to even low NO_2 concentrations and offers stability over time, ideal for continuous environmental monitoring.
- Low Power Consumption: Designed for low-power applications, making it suitable for battery-powered, remote, or portable air quality monitoring systems.
- Construction: It consists of a MEMS structure with a metal oxide semiconductor layer

that changes resistance in response to NO_2 gas levels.

MiCS-5524: CO Sensor

The MiCS-5524 is designed to detect Carbon Monoxide (CO), a colorless, odorless gas produced by incomplete combustion of fossil fuels. It's essential for indoor air quality systems, as high levels of CO can be life-threatening. Key specifications include:

- Measurement Range: 1 to 1000 ppm, covering typical CO exposure scenarios in both residential and industrial settings.
- Fast Response Time: This sensor reacts quickly to changes in CO concentration, which is crucial for real-time alerts in safety applications.
- Temperature and Humidity Compensation: The sensor's design allows it to perform reliably in various environmental conditions, ensuring accurate readings across temperature and humidity fluctuations.
- Low Power Draw and Durability: Like the MiCS-2714, this sensor is energy-efficient and suited for long-term, low-maintenance applications.

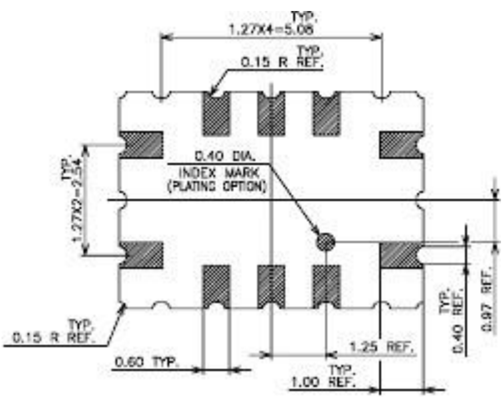

FIG :4.8 MICS-2714

The MiCS-2714 and MiCS-5524 are sensitive, compact metal oxide gas sensors widely used for air quality monitoring, particularly in urban and industrial environments. The MiCS-2714 specializes in detecting Nitrogen Dioxide (NO_2) from sources like vehicle exhaust and industrial emissions, with a measurement range of 0.05 to 5 ppm, ideal for capturing urban smog levels.

It offers high sensitivity, stability, and low power consumption, making it suitable for portable or remote environmental monitoring systems. On the other hand, the MiCS-5524 detects Carbon Monoxide (CO) in the range of 1 to 1000 ppm, ideal for indoor air quality and safety applications since CO is hazardous in confined spaces. Known for its quick response time and resilience in various temperature and humidity conditions, the MiCS- 5524 provides real-time, reliable CO monitoring. Both sensors are used in applications like smart city air quality networks, industrial safety systems, and IoT-based pollution tracking, offering low-power, durable, and efficient solutions for continuous monitoring of these critical pollutants.

INTRODUCTION TO AQI/AQS SENSOR

An AQI (Air Quality Index) sensor is a device that measures various air pollutants to determine overall air quality. AQI sensors monitor pollutants like Particulate Matter (PM2.5 and PM10),Carbon Monoxide (CO),Nitrogen Dioxide (NO_2),Ozone (O_3),Sulfur Dioxide (SO_2), and sometimes Volatile Organic Compounds (VOCs).

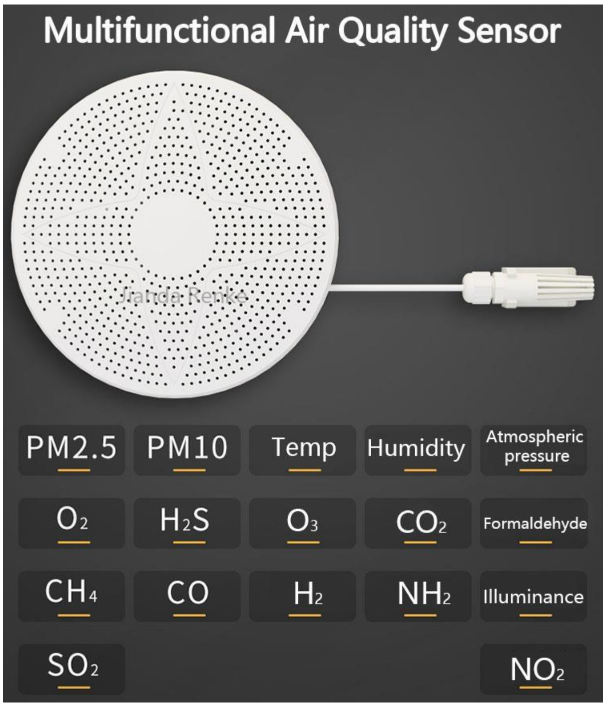

FIG:4.9 AQI SENSSOR

These pollutants impact respiratory health, and AQI sensors aggregate their readings to provide a standardized index value representing air quality on a scale (typically 0-500), with lower values indicating healthier air and higher values indicating more hazardous conditions. AQI sensors are commonly used in environmental monitoring systems, personal air quality devices, and smart city applications to help individuals and authorities monitor air pollution levels in real-time, take preventive actions, and implement policies for public health and environmental safety.

An AQI (Air Quality Index) sensor is designed to provide a comprehensive snapshot of air quality by tracking multiple pollutants and synthesizing these readings into a single, easy- to-understand AQI value. Modern AQI sensors often include wireless connectivity for real- time data transmission to cloud platforms, mobile apps, or centralized databases. This connectivity helps generate dynamic air quality maps, supports predictive analysis, and can trigger health alerts.

Introduction to Moinitoring Display

A monitoring display for air pollution is an interface or system that visually presents data collected from various sensors tracking air quality. These displays show real-time readings of pollutants like PM2.5, PM10, NO_2, CO, O_3, and other harmful substances in the air, often using graphs, numerical values, and color-coded indicators. The main purpose is to give immediate feedback to users about the current air quality, allowing them to take timely actions if pollution levels exceed safe thresholds. This can be critical for public health, especially in urban environments or areas with high industrial activity.

FIG:4.10 Display

These displays typically use Air Quality Index (AQI) values to simplify complex data into an easy-to-understand scale. The AQI is divided into categories, from "Good" to "Hazardous," each associated with a color (green to maroon), providing a quick visual cue for users to assess the air quality. The display might also show individual pollutant concentrations, trend graphs, and alerts for high pollutant levels. Advanced systems often feature touchscreens, interactive interfaces, and real-time updates from remote sensors, offering users the ability to interact with the data and explore detailed information about air pollution sources and effects.For public air quality monitoring systems, displays are usually installed in high-traffic areas like parks, schools, government buildings, or transportation hubs. These systems serve the dual purpose of informing the public about local air quality and encouraging healthier behaviors, such as reducing outdoor activities when air pollution is high. In more advanced or commercial settings, these displays may be integrated into broader environmental monitoring systems.

LCD/DISPLAY:

Liquid Crystal Display also called as LCD is very helpful in providing user interface as well as for debugging purpose. The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers

Pin Description:

Fig 4.11 Pin Description

LED'S

A light-emitting diode (LED) Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device. Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared (IR) light. Infrared LEDs are used in remote-control circuits, such as those used with a wide variety of consumer electronics.

The first visible-light LEDs were of low intensity and limited to red. Modern LEDs are available in visible, ultraviolet (UV), and infrared wavelengths, with high light output. Early LEDs were often used as indicator lamps, replacing small incandescent bulbs, and in seven-segment displays. Recent developments have produced high-output white light LEDs suitable for room and outdoor area lighting. LEDs have led to new displays and sensors, while their high switching rates are useful in advanced communications technology. LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. LEDs are used in applications as diverse as aviation lighting, fairy lights, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper, horticultural grow lights, and medical devices. Unlike a laser, the light emitted from an LED is neither spectrally coherent nor even highly monochromatic. However, its spectrum is sufficiently narrow that it appears to the human eye as a pure (saturated) color Also unlike most lasers, its radiation is not spatially coherent, so it cannot approach the very high brightnesses characteristic of lasers.

The display might also show individual pollutant concentrations, trend graphs, and alerts for high pollutant levels. Advanced systems often feature touchscreens, interactive interfaces, and real-time updates from remote sensors, offering users the ability to interact with the data and explore detailed information about air pollution sources and effects.

For public air quality monitoring systems, displays are usually installed in high-traffic areas like parks, schools, government buildings, or transportation hubs. These systems serve the dual purpose of informing the public about local air quality and encouraging healthier behaviors, such as reducing outdoor activities when air pollution is high. In more advanced or commercial settings, these displays may be integrated into broader environmental monitoring systems, providing data analytics, predictions.In addition to providing immediate alerts, buzzers can be programmed to emit different sound patterns or frequencies based on the severity of the air pollution. For example, a low-pitched beep might indicate moderate pollution levels, while a continuous or high-pitched sound could signal hazardous conditions.

Introduction to GSM Module

A **GSM module** (Global System for Mobile Communications) is a hardware component that allows electronic devices to communicate over a mobile network. It works by sending and receiving data via SMS (Short Message Service), calls, or even GPRS (General Packet Radio Service) for internet access. Commonly used in IoT (Internet of Things) applications, a GSM module can enable remote control, data transmission, and notification systems. It connects to a microcontroller or a microprocessor, and its primary function is to provide wireless communication capabilities, which are especially useful in applications where wired communication is not feasible or practical. Commonly used in IoT (Internet of Things) applications, a GSM module can enable remote control, data transmission, and notification systems. It connects to a microcontroller or a microprocessor, and its primary function is to provide wireless communication capabilities, which are especially useful in applications where wired communication is not feasible or practical.

GSM modules typically require a SIM card for network access, and their ease of integration with microcontrollers like Arduino and Raspberry Pi makes them an ideal solution for remote monitoring systems, home automation, security, and emergency alert applications.

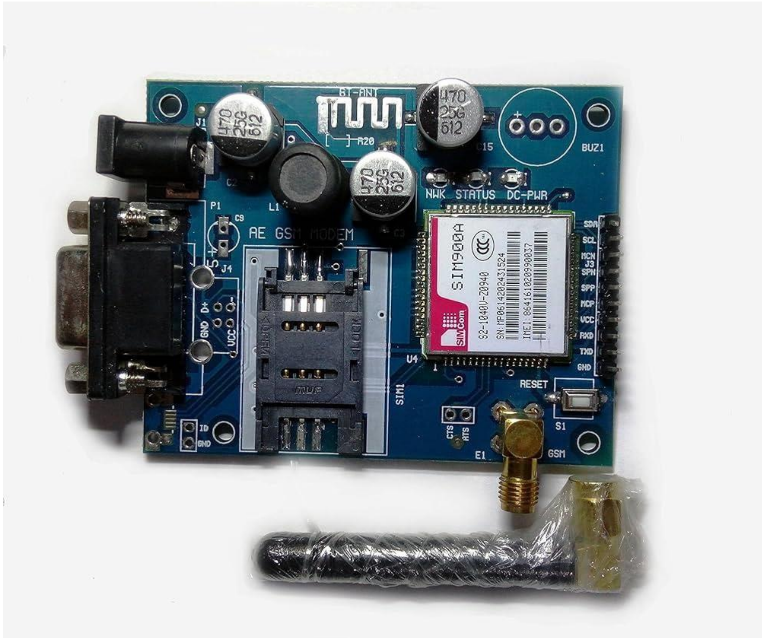

FIG:4.11 GSM MODULE

In air pollution monitoring systems, the GSM module is used to send real-time alerts, such as warnings about high pollution levels, to users or administrators via SMS. For example, if the pollution levels exceed a predefined threshold, the GSM module can send a text message to designated phone numbers, informing them of the situation and prompting actions like activating ventilation systems or evacuating the area. Additionally, the GSM module can be programmed to send periodic updates on air quality readings to remote monitoring centers, allowing for continuous data collection and monitoring, even without internet connectivity. The **SIM800** and **SIM900** modules are some of the most popular GSM modules, providing basic SMS and voice call capabilities. More advanced models, like those supporting GPRS or 3G/4G LTE networks, allow for faster data transmission, which is essential for applications that require real-time data streaming or internet-based services. GSM modules typically require a **SIM card** for network access, and their ease of integration with microcontrollers like Arduino and Raspberry Pi makes them an ideal solution for remote monitoring systems, home automation, security, and emergency alert applications.

4.6 INTRODUCTION TO BUZZER

A buzzer is an electronic component that emits sound signals to indicate an alert or notification, commonly used in devices that require audible feedback. In air pollution monitoring systems, a buzzer is often employed as an alert mechanism to notify users when pollutant levels exceed safe thresholds. When the system detects that pollution parameters such as PM2.5, CO, NO_2, or AQI levels go beyond the preset limits, the buzzer activates to provide an audible warning. This helps users take timely actions to mitigate exposure to harmful air quality.

FIG: 4.12 Buzzer

Buzzers can come in two main types: active and passive. Active buzzers have a built-in oscillator and only require a DC voltage input to generate sound, making them simpler to use in most electronic applications. On the other hand, passive buzzers require an external frequency signal to produce sound, offering more control over the pitch and tone of the sound. In air pollution monitoring, active buzzers are more commonly used as they can easily provide a consistent warning without the need for complex circuitry.

The integration of a buzzer into air pollution monitoring systems enhances their functionality, providing a crucial real-time notification that complements visual displays like LED indicators or screens. These audible alerts are especially beneficial in environments with high ambient noise, where visual cues alone may not be sufficient. The buzzer's sound can prompt users to take action, such as leaving a polluted area or turning on air purifiers, ensuring that people are aware of unhealthy air conditions quickly and effectively.

In addition to providing immediate alerts, buzzers can be programmed to emit different sound patterns or frequencies based on the severity of the air pollution. For example, a low-pitched beep might indicate moderate pollution levels, while a continuous or high-pitched sound could signal hazardous conditions. This feature allows users to differentiate between different levels of danger without needing to check the display, improving the system's responsiveness. Moreover, buzzers can be integrated into automated systems, where they serve as part of a broader alert network, triggering other actions like activating fans, closing ventilation systems, or notifying emergency services if pollution levels reach critical thresholds. This makes buzzers a simple but essential component in ensuring user safety in environments affected by poor air quality

4.7 INTRODUCTION TO POWER SUPPLY

A power supply is a critical component of a smart air pollutants monitoring system using IoT technologies, as it ensures reliable operation of all sensors, communication modules, and processing units. Depending on the application, the power supply must meet the specific voltage and current requirements of the devices used, such as MQ gas sensors, PM sensors, microcontrollers (e.g., Arduino, Raspberry Pi), GSM modules, and other components like buzzers and displays. In many IoT-based systems, power efficiency is a key concern since devices may need to run continuously or operate in remote locations without easy access to a direct power source.

For low-power applications, such as remote or battery-powered pollution monitors, a battery- powered supply is often used, with options like Li-ion or Li-poly batteries providing long- lasting power. These systems may also integrate solar panels for renewable energy, allowing the monitoring system to operate autonomously for extended periods, especially in outdoor environments. Solar-powered systems are especially useful in areas with frequent sunlight, as they can recharge the battery, making the system more sustainable and reducing maintenance costs. A low-power microcontroller, like an ESP32 or Arduino Nano, can also help extend battery life, as these devices are designed for minimal power consumption while still offering adequate processing capabilities for IoT tasks.

For more stationary or industrial-grade systems, where constant power availability is a concern, an AC-to-DC power supply can be used to convert wall outlet electricity into the required voltage for the system. This type of power supply is typically used in larger installations, such as air quality monitoring stations in urban areas, industrial facilities, or smart city infrastructure. These setups may also include backup power systems, like UPS (uninterruptible power supply), to ensure the system continues to function in case of power outages, providing additional reliability for continuous operation.

In more advanced IoT monitoring systems, where multiple sensors and communication modules (like GSM or Wi-Fi modules) need to work simultaneously, it is essential to manage power distribution effectively. Using a regulated power supply ensures that each component receives a consistent and adequate power level to function optimally. Power management circuits such as voltage regulators, DC-DC converters, and battery management systems can be used to optimize power consumption, extend battery life, and prevent overloading or damage to sensitive components, ensuring the long-term performance and stability of the monitoring system.

Security Features:

- **Multi-Factor Authentication:** Combines QR scanning, OTP verification, and RFID authentication for enhanced security.
- **Dynamic OTP:** One-Time Passwords are dynamically generated, ensuring they cannot be reused or guessed.
- **Real-Time Alerts:** Node MCU enables real-time notifications, ensuring quick action in case of unauthorized access attempts.
- **Digital Log Maintenance:** All access events are logged digitally for future audits and security analysis.
- **Enhanced Security:** Multi-layered authentication prevents unauthorized access.
- **Efficient and Automated:** The system automates vehicle entry and exit, reducing manual intervention.
- **User-Friendly Interface:** LCD display provides clear instructions and status updates.
- **IoT Integration:** Node MCU enables real-time monitoring and remote management.
- **Scalable Design:** The system can be expanded with additional security features or integrated with other IoT devices.

This methodology offers a comprehensive, secure, and efficient approach to managing unknown vehicle access in residential complexes. By leveraging QR scanning, OTP authentication, RFID technology, and IoT integration with Node MCU, the system provides a robust solution that enhances security, streamlines visitor management, and minimizes human errors. The architecture ensures scalability and flexibility, making it suitable for modern residential communities that demand advanced security systems. This is the world of the Internet of Things (IOT). Internet of things common definition is defining as: Internet of things (IOT) is a network of physical objects.

Power management circuits such as voltage regulators, DC-DC converters, and battery management systems can be used to optimize power consumption, extend battery life, and prevent overloading

ACTIVITY DIAGRAM:

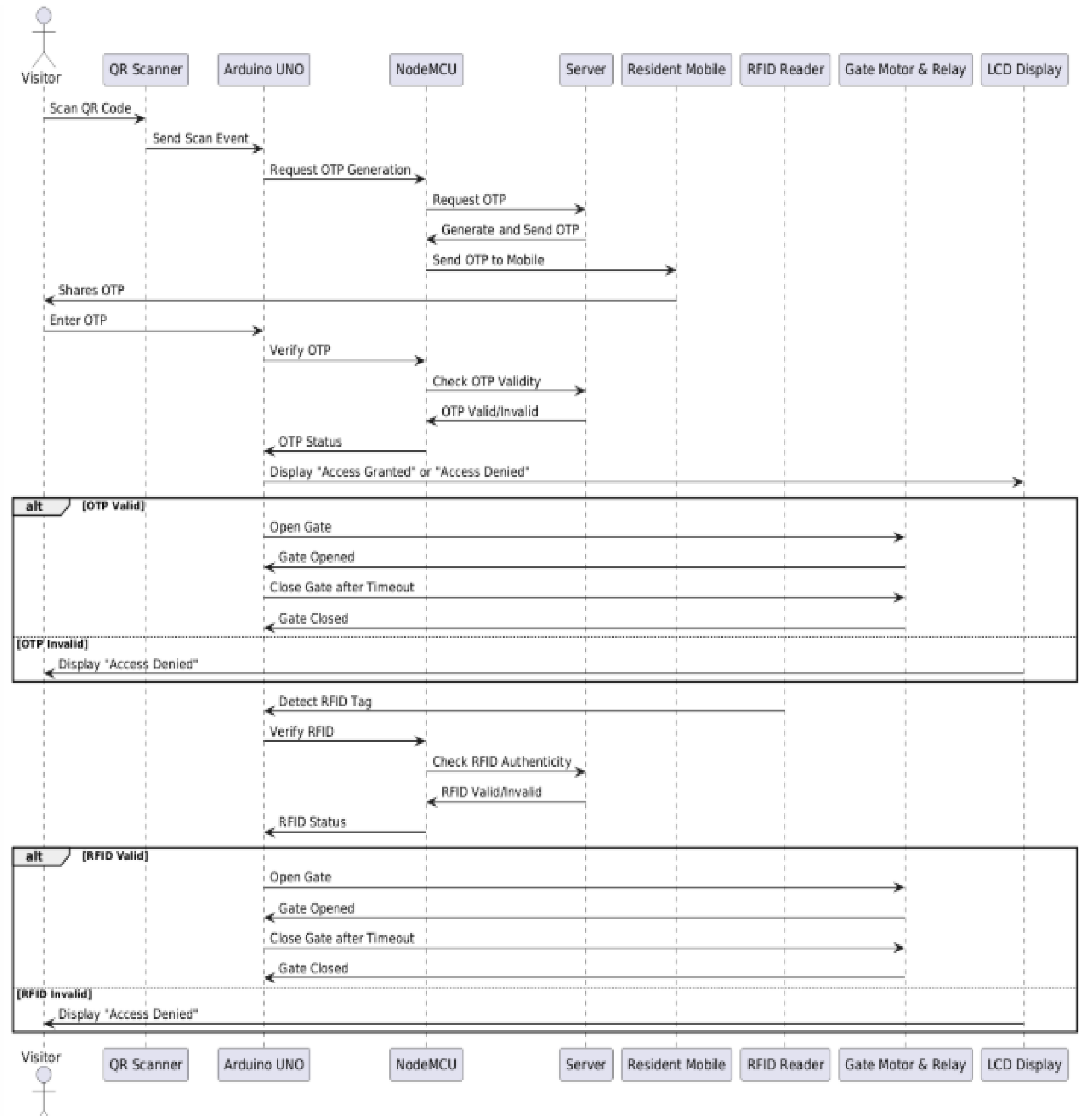

FIG 4.13 Activity Diagram

QR Code Authentication Flow:

1. The visitor scans the QR code at the gate entrance.

2. The QR scanner sends the event to Arduino UNO, which requests an OTP via NodeMCU.

3. Node MCU communicates with the server to generate and send the OTP to the authorized resident's mobile device.

4. The visitor enters the received OTP, which Arduino UNO verifies through NodeMCU and the server.

5. If the OTP is valid, Arduino opens the gate and then closes it automatically after a timeout. If the OTP is invalid, access is denied.

RFID Authentication Flow:

1. For registered residents, the RFID tag is detected by the RFID reader.
2. Arduino UNO verifies the tag's authenticity through NodeMCU and the server.
3. If valid, the gate opens automatically and closes after the vehicle passes through. If invalid, access is denied.

If authentication fails, the LCD displays "Access Denied" for the visitor's reference.

Sequence Diagram :

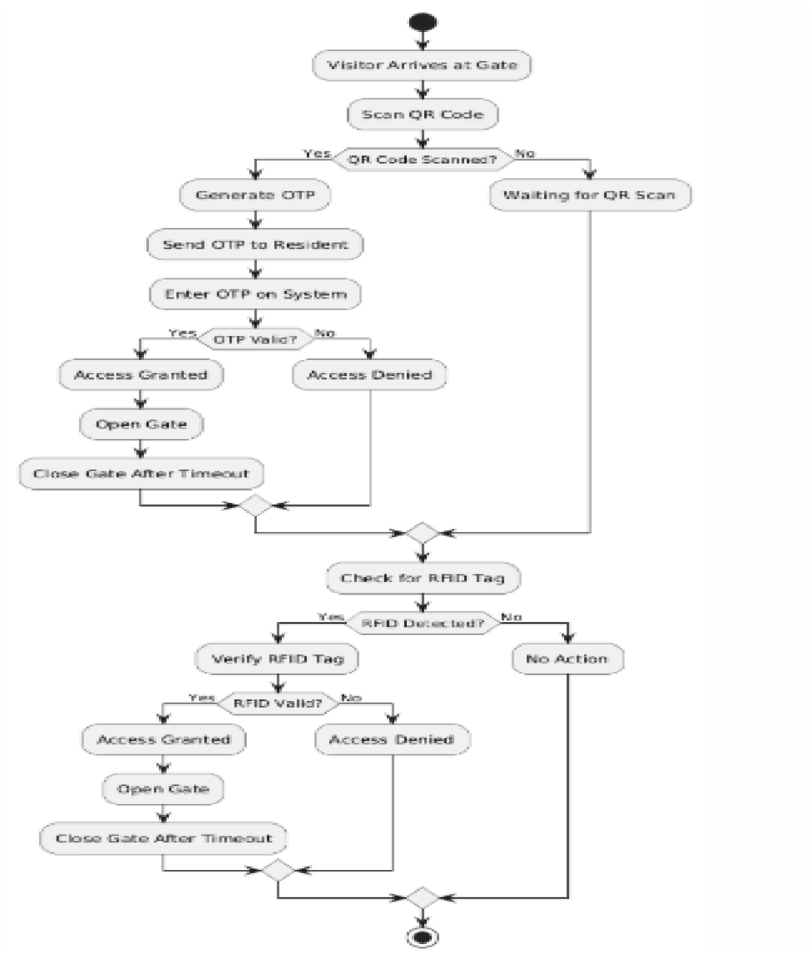

FIG 4.14 Sequence Diagram

CHAPTER-5

IOT (INTERNET OF THINGS)

5.1 INTERNET OF THINGS-IOT

The IOT concept was coined by a member of the Radio Frequency Identification (RFID) development community in 1999, and it has recently become more relevant to the practical world largely because of the growth of mobile devices, embedded and ubiquitous communication, cloud computing and data analytics.

Imagine a world where billions of objects can sense, communicate and share information, all interconnected over public or private Internet Protocol (IP) networks. These interconnected objects have data regularly collected, analysed and used to initiate action, providing a wealth of intelligence for planning, management and decision making.

This is the world of the Internet of Things (IOT). Internet of things common definition is defining as: Internet of things (IOT) is a network of physical objects. The internet is not only a network of computers, but it has evolved into a network of device of all type and sizes, vehicles, smart phones, home appliances, toys, cameras, medical instruments and industrial systems, animals, people, buildings, all connected, all communicating & sharing information based on stipulated protocols in order to achieve smart reorganizations, positioning, tracing, safe & control & even personal real time online monitoring.

online upgrade, process control & administration We define IOT into three categories as below: Internet of things is an internet of three things People to people, People to machine /things, Things /machine to things /machine, Interacting through internet.

Internet of Things Vision:

Internet of Things (IoT) is a concept and a paradigm that considers pervasive presence in the environment of a variety of things/objects that through wireless and wired connections and unique addressing schemes are able to interact with each other and cooperate with other things/objects to create new applications/services and reach common goals.

In this context the research and development challenges to create a smart world are enormous. A world where the real, digital and the virtual are converging to create smart environments that make energy, transport, cities and many other areas more intelligent.

Internet of Things is refer to the general idea of things, especially everyday objects, that are readable, recognisable, locatable, addressable through information sensing device and/or controllable via the Internet, irrespective of the communication means (whether via RFID, wireless LAN, wide area networks, or other means). Everyday objects include not only the electronic devices we encounter or the products of higher technological development such as vehicles and equipment but things that we do not ordinarily think of as electronic at all - such as food , clothing ,chair, animal, tree, water etc. Internet of Things is a new revolution of the Internet. Objects make themselves recognizable and they obtain intelligence by making or enabling context related decisions thanks to the fact that they can communicate information about themselves. They can access information that has been aggregated by other things, or they can be components of complex services.

FIG:5.1 Internet of Things

This transformation is concomitant with the emergence of cloud computing capabilities and the transition of the Internet towards IPv6 with an almost unlimited addressing capacity. The goal of the Internet of Things is to enable things to be connected anytime, anyplace, with anything and anyone ideally using any path/network and any service.

ENABLING TECHNOLOGIES FOR IOT

Internet of things (IoT) is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies. With the Internet of Things the communication is extended via Internet to all the things that surround us. The Internet of Things is much more than machine to machine communication, wireless sensor networks, sensor networks , 2G/3G/4G,GSM,GPRS,RFID, WI-FI, GPS, microcontroller, microprocessor etc. These are considered as being the enabling technologies that make "Internet of Things" applications possible. Enabling technologies for the Internet of Things are considered in and can be grouped into three categories: (1) technologies that enable "things" to acquire contextual information, (2) technologies that enable "things" to process contextual information, and (3) technologies to improve security and privacy. The first two categories can be jointly understood as functional building blocks required building "intelligence" into "things", which are indeed the features that differentiate the IoT from the usual Internet. The third category is not a functional but rather a de facto requirement, without which the penetration of the IoT would be severely reduced The Internet of Things is not a single technology, but it is a mixture of different hardware & software technology. The Internet of Things provides solutions based on the integration of information technology, which refers to hardware and software used to store, retrieve, and process data and communications technology which includes electronic systems used for communication between individuals or groups. There is a heterogeneous mix of communication technologies, which need to be adapted in order to address the needs of IoT applications such as energy efficiency, speed, security, and reliability. In this context, it is possible that the level of diversity will be scaled to a number a manageable connectivity technologies that address the needs of the IoT applications, are adopted by the market, they have already proved to be serviceable, supported by a strong technology alliance. Examples of standards in these categories include wired and wireless technologies like Ethernet, WI-FI, Bluetooth, ZigBee, GSM, and GPRS. [1, 2] The key enabling technologies for the Internet of Things is presented.

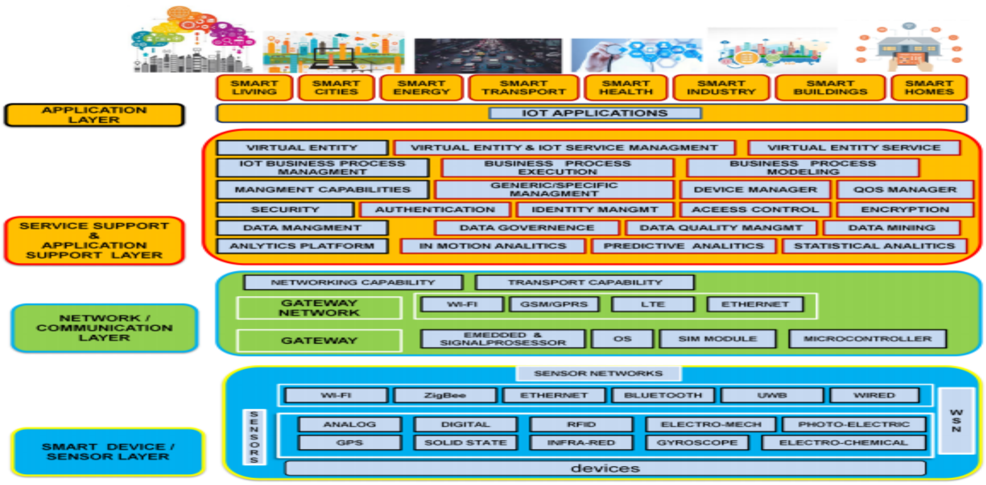

FIG:5.2 IOT ARCHITECTURE

CHAPTER-6

RESULT

The implementation of a QR scanner and OTP system for unknown vehicle access in an apartment complex ensures enhanced security and controlled entry. When an unidentified vehicle arrives, the system generates a unique QR code or sends an OTP to the concerned resident or an authorized individual. This process requires the resident to verify and approve the entry, thus confirming the vehicle's legitimacy. By combining QR code technology and OTP verification, unauthorized access is minimized.

Show this QR code to generate otp.

FIG:6.1 QR CODE

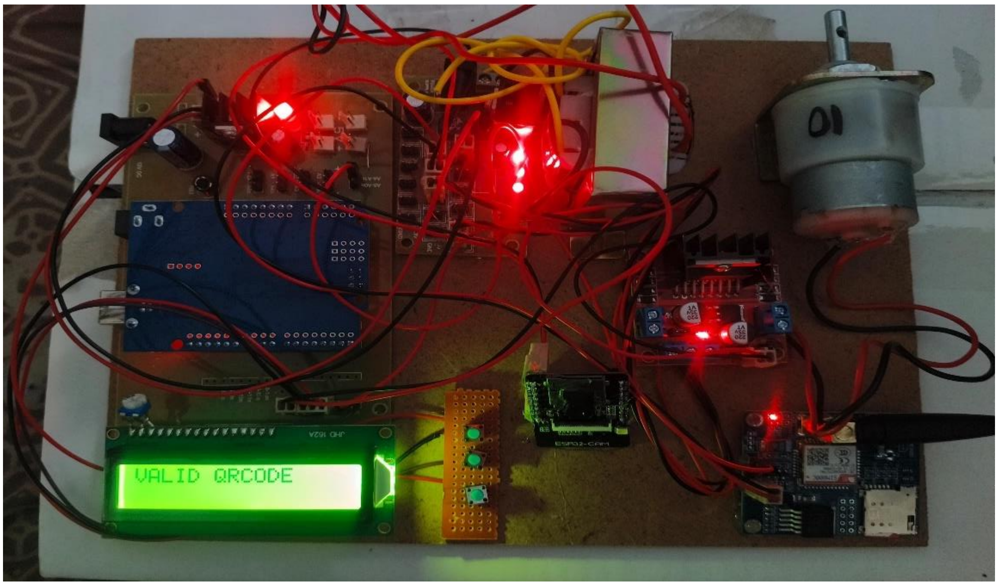

FIG:6.2 QR Code Verified

FIG: 6.3 OTP Sent to Mobile

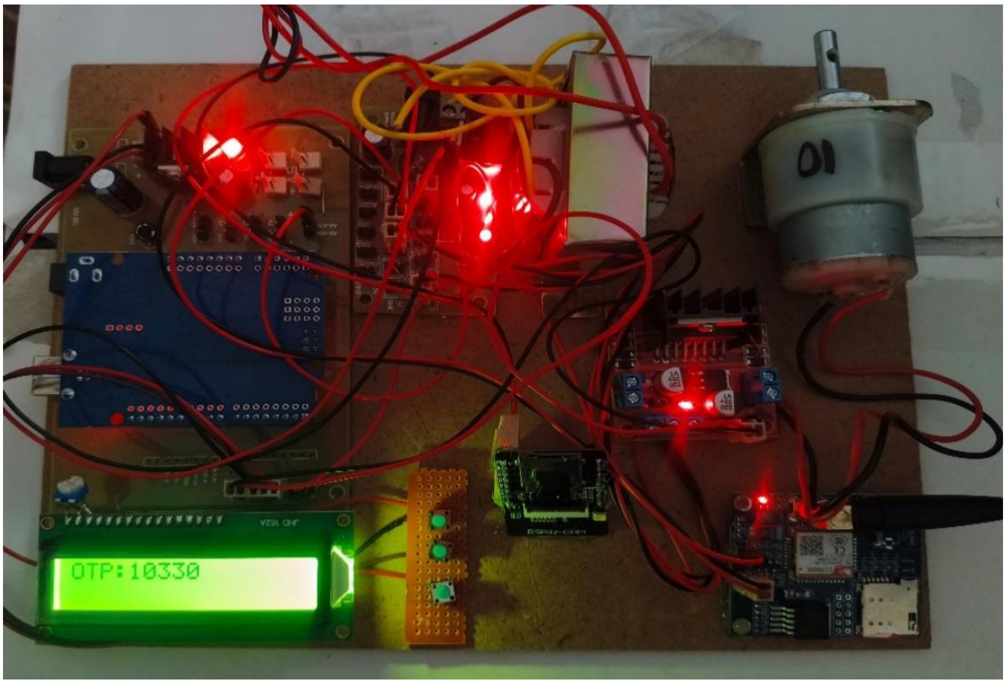

FIG:6.4 OTP Entered

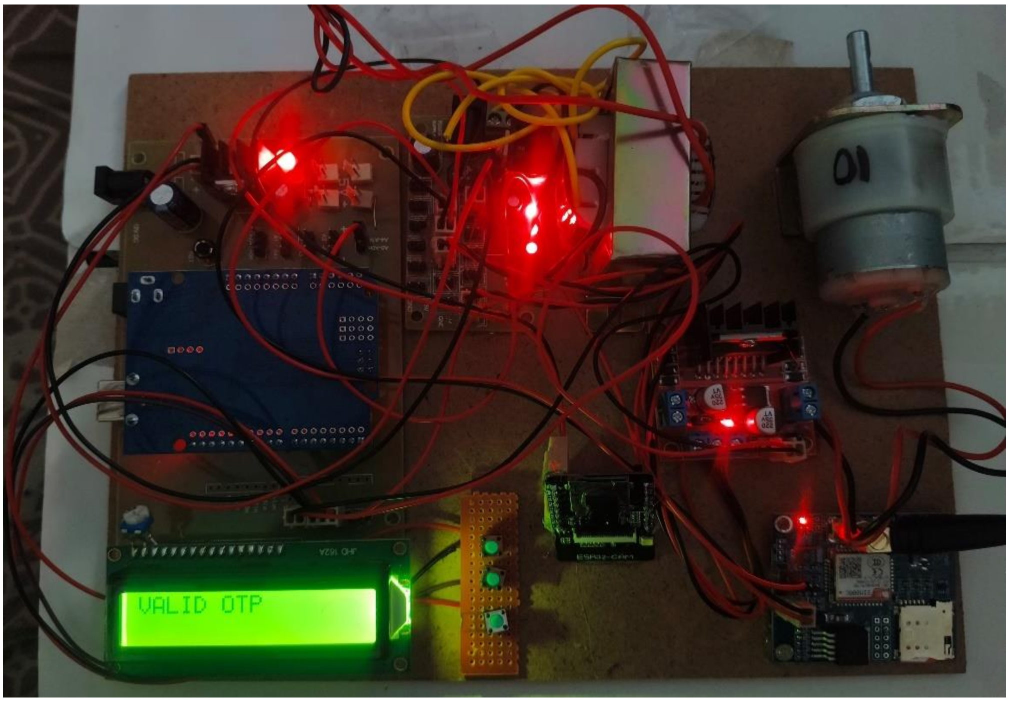

FIG:6.5 OTP Verified

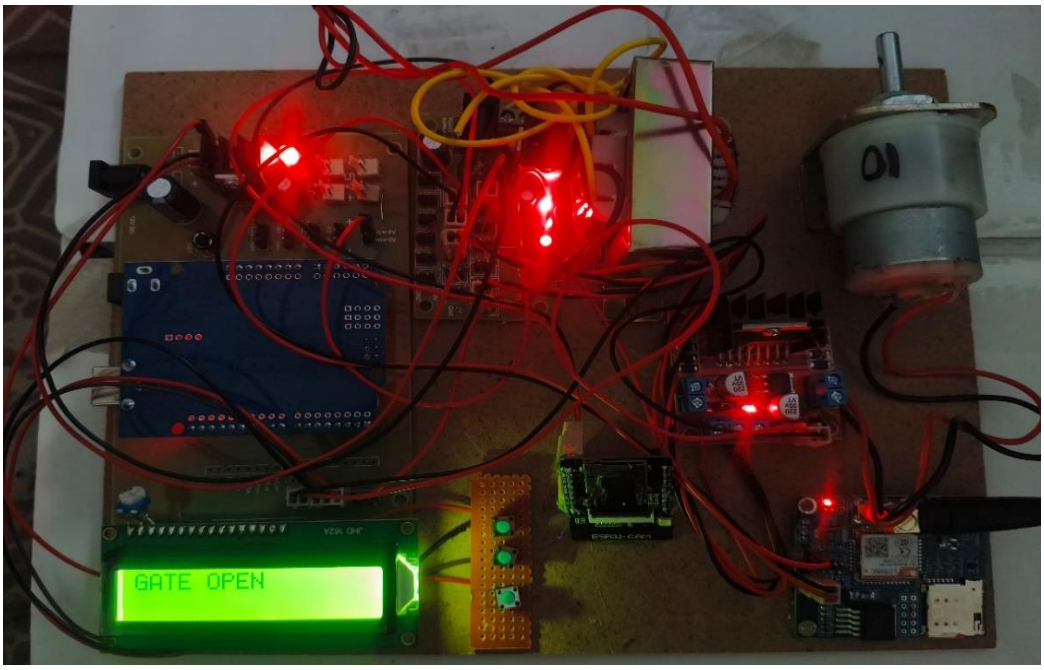

FIG:6.6 OTP Verified and Gate Opened

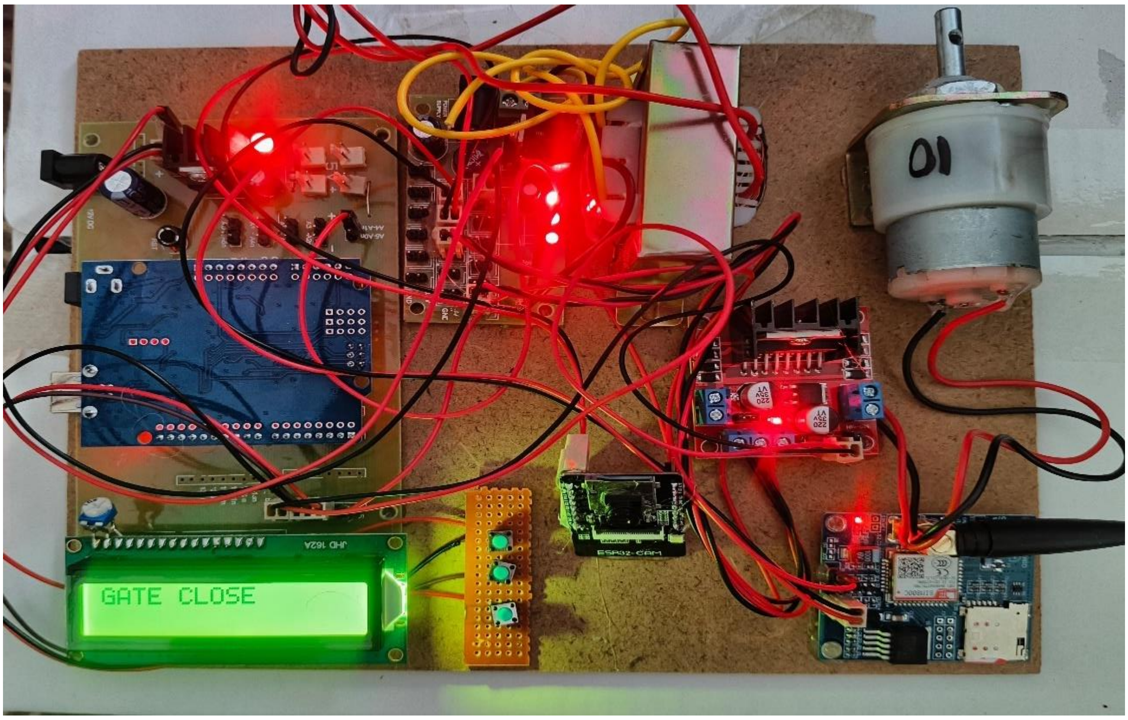

FIG:6.7 Gate Closed

CONCLUSION

The proposed smart access control system using QR scanner and OTP authentication, integrated with Arduino UNO, NodeMCU, and RFID technology, provides a robust and efficient solution for managing vehicle entry in residential complexes, such as apartments and gated communities. By implementing a multi-layered authentication process, the system enhances security, prevents unauthorized access, and eliminates the limitations of traditional methods like manual verification or fixed RFID tags. The use of IoT components, such as NodeMCU, enables real-time monitoring and remote management, making the system not only secure but also user-friendly and scalable. Additionally, the integration of an LCD display and automated gate control mechanism ensures seamless user interaction and operational efficiency. Overall, this solution offers a modern, reliable, and convenient way of managing vehicle access, significantly improving the security framework of residential complexes.

FUTURE SCOPE

The proposed smart access control system has significant potential for future enhancements to further improve security, convenience, and scalability. One promising direction is the integration of a mobile application, enabling residents and security personnel to monitor gate access, approve visitors, and receive real-time notifications. Advanced security features, such as facial recognition and biometric authentication, can be incorporated to provide an additional layer of verification. Furthermore, the system can benefit from cloud-based data management, allowing for secure storage of access logs and advanced analytics through machine learning algorithms to detect unusual activity patterns. Integrating voice command functionalities with popular smart assistants like Alexa or Google Assistant can enhance user experience by enabling hands-free operation. Additionally, the system can be expanded to support multiple gates within large residential complexes, ensuring scalability and adaptability. Energy efficiency can be achieved by incorporating solar-powered components for uninterrupted operation during power outages. Safety features, such as emergency override options and SOS alerts, can be integrated to enhance resident safety. Overall, the proposed system's flexible architecture allows for continuous improvements, making it a future-ready solution for modern smart communities.

REFERENCES

[1] Tarun Agarwal, (2015), "RFID based Automatic Door Lock System with Arduino", Retrieved 19 October 2017, from https://www.elprocus.com/automatic-door-lock-system-using rfid-and-arduino/.

[2].Ramalatha, M., Ramkumar, A. K., Selvaraj, S. and Suriyakanth, S, (2014), "RFID Based Personal Medical Data card for Toll Automation". Elysium Journal Engineering Research and Management. Volume1, ISSN: 2347-4408, pp 51-52.

[3].Umar Farooq, Mahmood ul Hasan, Muhammad Amar, Athar Hanif, and Muhammad Usman Asad, (2014), "RFID Based Security and Access Control System" IACSIT International Journal of Engineering and Technology, Vol. 6, No. 4, August 2014.

[4].Peter Adole1, Joseph M. Môm and Gabriel A. Igwue, (2016), "RFID Based Security Access Control System with GSM Technology", American Journal of Engineering Research (AJER) e-ISSN: 2320-0847 p-ISSN: 2320-0936 Volume-5, Issue-7, pp-236-242.

[5].Dejan Nedelkovski, (2017), "How RFID Works and How To Make an Arduino based RFID Door Lock - HowToMechatronics", Retrieved 15 October 2017, from

[6].http://howtomechatronics.com/tutorials/arduino/rfid-works-make-arduino-based-rfid door- lock

[7].Massimo Banzi, "Getting Started with Arduino" USE, Dale Dougherty, 2009

APPENDIX

```
#include <SoftwareSerial.h>
#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);
SoftwareSerial esp(A4, A5);

int m1 = 7;
int m2 = 6;
#define b1 5
#define b2 4
#define b3 3

unsigned char key[5];
unsigned int temp = 0;
signed int val = 0;
int OTP = 0, otp = 0;
String number = "9390608557";
int message = 0;

void regnumber() {
  /*String number="9390608557";
  int message=0;*/
  lcd.clear();
  lcd.print("AT");
  Serial.print("AT\r\n");
  delay(1000);
  lcd.clear();
  lcd.print("ATE0");
  Serial.print("ATE0\r\n");
  delay(1000);
  lcd.clear();
  lcd.print("AT+CMGF=1");
  Serial.print("AT+CMGF=1\r\n");
  delay(1000);
  lcd.clear();
```

```
    lcd.print("AT+CNMI=1,2,0,0");
    Serial.print("AT+CNMI=1,2,0,0\r\n");
    delay(1000);
    Serial.end();
    delay(10000);
    Serial.begin(9600);
    delay(1000);
    lcd.clear();
    lcd.print("PLZ STORE NUM");

back:
    while (Serial.available() > 0) {
        // String number="9390608557";
        // int message=0;
        String siva = Serial.readString();
        int len = siva.length();
        int ourdata = siva.indexOf("NUMBER:");
        lcd.clear();
        lcd.print("message received");
        delay(1000);
        lcd.clear();
        lcd.print("Ph.No:");
        for (int i = 7; i <= 17; i++) {
            number[i - 7] = siva[ourdata + i];
        }
        message = 1;
        lcd.print(number);
        delay(1000);
        Serial.end();
        delay(10000);
        Serial.begin(9600);
        delay(1000);
        lcd.clear();
        lcd.print("Sending SMS...");
        lcd.setCursor(0, 1);
        lcd.print("To  ");
        lcd.print(number);
```

```
        delay(1000);
        Serial.print("AT+CMGS=");
        Serial.print("");
        Serial.print(number);
        Serial.print("");
        Serial.print("\r\n");
        delay(1000);
        Serial.print("Welcome Message From Hardware Kit");
        delay(100);
        Serial.write(0x1A);
        delay(10000);
        Serial.end();
        delay(10000);
        Serial.begin(9600);
        delay(1000);
        lcd.clear();
        lcd.print(" SMS SENT.....");
        delay(1000);
        lcd.clear();
        lcd.print("waiting for input");
    }
    if (message == 0)
        goto back;
}

void setup() {
    pinMode(m1, OUTPUT);
    pinMode(m2, OUTPUT);
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    lcd.begin(16, 2);
    lcd.print("hello, world!");
    pinMode(b1, INPUT_PULLUP);
    pinMode(b2, INPUT_PULLUP);
    pinMode(b3, INPUT_PULLUP);
    lcd.begin(16, 2);
    delay(1000);
```

```
lcd.setCursor(0, 0);
lcd.print("PARKING ACCESSING");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("USING OTP");
delay(1000);
Serial.begin(9600);
delay(1000);
esp.begin(9600);
delay(1000);
lcd.clear();
lcd.print("AT");
Serial.print("AT\r\n");
delay(1000);
lcd.clear();
lcd.print("ATE0");
Serial.print("ATE0\r\n");
delay(1000);
lcd.clear();
lcd.print("AT+CMGF=1");
Serial.print("AT+CMGF=1\r\n");
delay(1000);
lcd.clear();
lcd.print("AT+CNMI=1,2,0,0");
Serial.print("AT+CNMI=1,2,0,0\r\n");
delay(1000);
lcd.clear();
lcd.print("Sending SMS...");
lcd.setCursor(0, 1);
lcd.print("To  ");
lcd.print(number);
delay(1000);
Serial.print("AT+CMGS=");
Serial.print("");
Serial.print(number);
Serial.print("");
Serial.print("\r\n");
```

```
    delay(1000);
    Serial.print("Welcome Message From Hardware Kit");
    delay(100);
    Serial.write(0x1A);
    delay(10000);
    lcd.clear();
    lcd.print("ESP32 CAM INPUT");
    delay(1000);
}

void loop() {
back:
    OTP = (millis() / 1000);
    delay(1000);
    if (OTP >= 99998) {
        OTP = 0;
    }
    while (esp.available()) {
        String siva = esp.readString();
        lcd.clear();
        lcd.print("ESP32 CAM:");
        lcd.print(siva);
        delay(1000);
        int ourdata = siva.indexOf("Payload:");
        lcd.clear();
        lcd.print("Received:");
        for (int i = 9; i < 14; i++) {
            lcd.print(siva[ourdata + i]);
        }
        delay(1000);
        if (siva.substring(ourdata + 9, ourdata + 14) == "12345") {
            lcd.clear();
            lcd.print("VALID QRCODE");
            delay(3000);
            lcd.setCursor(0, 1);
            lcd.print("SENDING OTP");
             otp = 9999 + OTP;
```

```
        lcd.clear();
        lcd.print("OTP:");
        lcd.print(otp);
        delay(2000);
        lcd.clear();
        lcd.print("SENDING OTP...");
        delay(1000);
        Serial.print("AT+CMGS=");
        Serial.print("");
        Serial.print(number);
        Serial.print("");
        Serial.print("\r\n");
        delay(1000);
        Serial.print("OTP FROM ARDUINO TO OPEN DOOR:");
        Serial.print(String(otp));
        delay(100);
        Serial.write(0x1A);
        delay(10000);
        lcd.clear();
        lcd.print(" SMS SENT.....");
        delay(1000);
      }
    }
}
```