SMART DOOR LOCK SYSTEM

Thesis submitted in partial fulfilment of
the Requirements for the Degree of

MASTER OF TECHNOLOGY

IN

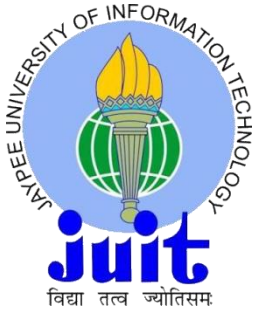
ELECTRONICS AND COMMUNICATION ENGINEERING

By

Kshitij Verma (212054)

UNDER THE GUIDANCE OF

Dr. Emjee Puthooran



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
May 2023

TABLE OF CONTENT

DECLARATION

I hereby declare that the work reported in the M.Tech Dissertation Report entitled **" Smart Door Lock System "** submitted at the **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of our work carried out under the supervision of **Dr. Emjee Puthooran**. We have not submitted this work elsewhere for any other degree or diploma.

Kshitij verma
Roll: 212054

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Emjee Puthooran	Prof. Dr. Rajiv Kumar
(Associate Professor, ECE, JUIT)	(Professor and Head of Department of ECE, JUIT)
Date:	Date:

ACKNOWLEDGEMENT

Above all, I thank God, the Almighty for providing me with this opportunity and granting me the capability to proceed successfully. The present work will remain incomplete unless I express my feeling of gratitude towards several persons who delightfully co-operated with me in the process of the work.

It is my privilege to express my sincere gratitude to Prof. Rajeev Kumar, Head of the Department of Electronics and Communications Engineering Department, Jaypee University of Information Technology, Waknaghat, India, for their inspiration and motivation during my study.

I solemnly express my earnest and humble thanks to my supervisor Dr. Emjee Puthooran, Department of Electronics and Communication Engineering, JUIT Waknaghat, for the encouragement, guidance, and valuable suggestion to accomplish this thesis work.

Finally, I am thankful to my family members, friends, and all who have contributed to this thesis work directly or indirectly in completing my present study.

Kshitij Verma

MTech (ECE-IoT)

(Roll Number:212054)

LIST OF ABBREVIATIONS

S.No.	Abbreviation	Full Form
1	AWS	Amazon Web Services
2.	IoT	Internet of Things.
3.	MQTT	Message Queue Telemetry Transport
4.	Wi-Fi	Wireless-Fidelity
5.	IPV	Internet Protocol Version
6.	OpenSSL	Open Secure Socket layer.
7.	OpenCV	Open Computer Vision Library
8.	SOC	System On Chip
9.	ESP	Espressife System
10.	GND	Ground
11.	ADC	Analog to Digital Converter
12.	DAC	Digital to Analog Convertor
13.	Rx	Receiver
14.	Tx	Transmitter
15.	I2C	Inter-Integrated Circuit
16.	SPI	Serial Peripheral Interface
17.	IO	Input/Output
18	SDL	System Door Lock

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The term "IoT" is gaining increasing popularity, with IoT devices proving their usefulness in variousscenarios. This project focuses on a smart lock system that can be controlled remotely through smartphones, offering convenience and accessibility. By capturing images of individuals at the doorusing an ESP32 device and comparing them with a database stored on Amazon Web Services (AWS),authorized access can be granted, enabling the door to be remotely unlocked. This solution is particularly beneficial for elderly and disabled individuals who may struggle with physical access. Additionally, the system provides real-time video coverage and notifications, allowing users to identify and interact with visitors. With its cost-effective implementation and user-friendly interface, this project showcases the potential of IoT in enhancing security and improving the lives of individuals through smart home automation.

CHAPTER – 1

INTRODUCTION

A smart home system refers to a network of interconnected devices and appliances that are integrated with advanced technologies to automate and enhance various aspects of home management and control. By leveraging the power of the Internet of Things (IoT), smart home systems enable homeowners to remotely monitor and control their household devices, such as lighting, heating, security systems, and entertainment systems, through a centralized control interface. These systems offer a range of benefits, including increased energy efficiency, enhanced security, improved convenience, and personalized experiences. Smart home systems utilize sensors, actuators, and communication technologies to gather data, analyze patterns, and make intelligent decisions, creating a seamless and interconnected living environment. Through the integration of technologies like voice assistants, mobile applications, and cloud services, smart home systems provide homeowners with the ability to customize and automate their home environment according to their preferences and needs. With the rapid advancements in IoT and home automation technologies, smart home systems are transforming the way we interact with our living spaces, making our homes more efficient, secure, and comfortable.

In this report, two different projects will be discussed.

I. Smart Door Lock System using ESP32-CAM and telegram.
II. Smart Door Lock System using Amazon Web Service.

I. Smart Door Lock System using ESP32-CAM and telegram.

The first project focuses on the development of a smart door lock system utilizing the ESP32-CAM module and Telegram for user interaction. The aim is to enhance the security and convenience of conventional door lock systems by integrating IoT technology and a user-friendly mobile application.

The ESP32-CAM module serves as the central component, providing functionalities such as video streaming and image capture. By establishing an internet connection, users can remotely monitor andcontrol the door lock system through the Telegram app.

Telegram acts as the interface between the user and the smart door lock system, enabling real-time

notifications, door locking/unlocking, and live video streaming. The system utilizes the MQTT protocol for efficient and reliable communication between the ESP32-CAM and Telegram.

This report will delve into the research work conducted, the hardware components utilized, the designand implementation process, and the achieved results. Furthermore, it will explore the significance of IoT integration, the protocols employed, and the overall impact of the smart door lock system.

The subsequent sections will provide a comprehensive overview of the project, commencing with a literature review to gain insights into existing technologies and implementations relevant to smart door lock systems.

II. Smart Door Lock System using Amazon Web Service.

Our second project focuses on leveraging AWS (Amazon Web Services) as a bridge between users and Telegram, along with utilizing AWS's face detection and recognition capabilities. The objective

is to enhance the functionality of a smart door lock system by incorporating advanced features such as person recognition and real-time notifications.

Traditionally, door lock systems have relied on manual operations or basic remote-control mechanisms. Our project aims to elevate the security and convenience of such systems by integrating AWS's powerful services. By leveraging AWS's extensive cloud infrastructure, we can seamlessly connect user interactions through Telegram with the smart door lock system. The project consists of four key phases. In the first phase, we establish a connection between the ESP32-CAM module and AWS, enabling data transfer between the devices. In the second phase, wefocus on bidirectional data transfer between the ESP32-CAM and AWS, enabling seamless communication and control.

The third phase revolves around utilizing AWS's face detection and recognition capabilities. By integrating these features, the system can automatically identify and recognize individuals at the door,providing real-time notifications to users via Telegram. If an unrecognized person is detected, the system sends a photo to the user, enabling them to make informed decisions regarding door access.In the final phase, we integrate Telegram to provide user-friendly access and control of the smart doorlock system. Users can interact with the system through the Telegram app, granting or denying access remotely.This report will delve into the research work conducted, hardware equipment used, the design and implementation process, and conclude with an overview of the achieved results and future possibilities.

1.1Background

The Internet of Things (IoT) has emerged as a revolutionary technology that enables the connection of various devices and objects to the Internet. This connectivity allows them to communicate, exchange data, and perform intelligent actions. Within the realm of home automation and security, IoT presents significant opportunities for the development of smart and secure systems. In the context of home automation and security, IoT offers tremendous potential for creating smart and secure systems. Traditional door lock systems often rely on physical keys, which can be lost or stolen, and lack advanced features like remote access and user identification. These limitations pose security risks and inconvenience for homeowners. Therefore, there is a need to develop an IoT-based smart door lock system that integrates advanced technologies to overcome these challenges and provide a higher level of security and convenience.

The IoT has gained significant attention and adoption in recent years due to its potential to revolutionize various industries and domains. In the context of home automation, the IoT offers tremendous opportunities for creating smart home systems that enhance convenience, comfort, and security for homeowners.

A smart home system leverages IoT technologies to connect and control various devices within a home, such as lighting systems, thermostats, security cameras, door locks, and appliances. These devices are equipped with sensors and connected to a central hub or gateway, allowing homeowners to remotely monitor and manage them through a smartphone or other internet-connected devices.

The benefits of a smart home system are manifold. It offers homeowners the ability to automate routine tasks, such as adjusting lighting and temperature settings, scheduling appliance usage, and monitoring energy consumption. Additionally, smart home systems provide enhanced security features, such as remote surveillance and access control, allowing homeowners to monitor their homes and receive real-time alerts in case of any unusual activities.

Moreover, smart home systems contribute to energy efficiency and sustainability by optimizing resource consumption and reducing waste. By intelligently managing energy usage and providing insights into energy patterns, homeowners can make informed decisions to conserve energy and lowertheir utility bills.

The proliferation of IoT devices and the availability of robust communication protocols and cloud-based platforms have made smart home systems more accessible and affordable for homeowners.

With advancements in artificial intelligence and machine learning, smart home systems can learn andadapt to user preferences, creating personalized and intuitive experiences.

As the demand for connected and intelligent homes continues to grow, the integration of IoT technologies in the home environment holds immense potential. It not only enhances the quality of life for homeowners but also opens up new possibilities for energy management, healthcare monitoring, and assisted living for the elderly and individuals with disabilities.

In this report, we delve into the design and implementation of a smart home system that leverages IoT technologies. We explore the integration of devices, the utilization of data analytics and machinelearning algorithms, and the interaction with users through intuitive interfaces. Through this project,we aim to showcase the capabilities of IoT in transforming traditional homes into intelligent, connected spaces that offer enhanced comfort, convenience, and security for homeowners.

1.2 Problem Statement

The project aims to address the challenges and limitations faced in traditional home security systemsand access control mechanisms. These existing systems often lack advanced features, integration capabilities, and remote accessibility. Some key problems identified are:

Limited Remote Access: Traditional home security systems require physical presence or proximity to interact with them. This restricts the ability to monitor and control the system remotely, posing challenges for users who are away from home or need to grant access to others.

Lack of Smart Integration: Conventional access control mechanisms often operate independently, lacking integration with other smart devices or platforms. This hinders the seamless integration of security systems into a broader home automation ecosystem.

Inadequate User Interaction: The user interaction with existing systems is often limited to physical keypads, which can be cumbersome and less intuitive. This reduces the overall user experience and convenience.

Limited User Identification: Many conventional systems rely solely on traditional keys or passcodesfor user identification, which can be vulnerable to unauthorized access. A more robust and reliable user identification and authentication method is required.

By addressing these challenges, the project aims to develop an advanced smart home security and access control system that leverages the power of IoT, cloud computing, and modern technologies

toprovide enhanced security, convenience, and remote accessibility.

1.3 Objective

1.3.1 Investigate the Potential of IoT in Smart Home Security

The first objective of the project is to conduct a comprehensive investigation into the potential of IoTin enhancing smart home security. This involves studying the existing literature and research on IoT applications in the field of home security and identifying the key challenges and opportunities. By analyzing various IoT technologies, protocols, and architectures, the project aims to determine the most suitable approach for implementing a secure and efficient smart home security system.

1.3.2 Design an Integrated System Architecture

The project aims to design an integrated system architecture that seamlessly connects the ESP32-CAM module, cloud services, and the Telegram app. This involves designing the communication protocols, data flow, and interaction between different components of the system. The architecture will be designed to ensure compatibility, scalability, and reliability, allowing for the smooth transmission of data, commands, and alerts between the various system elements.

1.3.3 Develop ESP32-CAM and AWS Integration

A key objective is to develop the integration between the ESP32-CAM module and the AWS cloud platform. This involves configuring the ESP32-CAM module to establish a secure connection with AWS IoT Core using MQTT. The project aims to implement bidirectional communication, enablingthe module to send data such as video streams and sensor readings to AWS and receive commands or alerts from AWS. The integration will leverage AWS IoT services such as AWS IoT Core, AWS S3, and AWS Lambda for seamless data transmission and processing.

1.3.4 Implement Face Detection and Recognition

The project aims to implement face detection and recognition capabilities using AWS services. Thisinvolves utilizing AWS Recognition, a powerful deep learning-based service, to detect and recognizefaces in the captured video streams. The system will be trained to identify registered individuals andraise alerts if an unrecognized face is detected. The objective is to achieve accurate and reliable facedetection and recognition performance, enhancing the security and access control features of the smart home system.

1.3.5 Integrate Telegram App for User Interaction

An important objective is to integrate the Telegram app into the smart home security system for userinteraction and control. This involves developing a Telegram bot interface the user and system. Userswill be able to receive real-time notifications, view live video feeds, and remotely control the door lock through the Telegram app. The integration will leverage the Telegram Bot API and secure communication protocols to ensure seamless and secure interaction between the user and the smart home system.

1.3.6 Implement Secure Data Transmission

One of the primary objectives of the project is to ensure the secure transmission of data between theESP32-CAM module, cloud services, and the Telegram bot. This involves implementing robust encryption protocols and techniques to safeguard sensitive information and prevent unauthorized access. By utilizing secure communication channels such as HTTPS and employing encryption algorithms like SSL/TLS.

1.3.7 Create an Alert System

Another objective is to develop an alert system that can detect and notify users of any suspicious activities or security breaches. The system will be designed to monitor events such as unauthorized access attempts, tampering with the door lock, or unusual behavior around the premises. When suchevents occur, the system will generate alerts and send them to users through the Telegram app. This real-time notification feature enables users to take immediate action and enhances the overall securityof the smart home system.

1.3.8 Provide Real-time Monitoring

The project aims to enable real-time monitoring of the smart home security system. Users will havethe capability to remotely view the live video feed from the ESP32-CAM module using the Telegram app. This functionality allows users to monitor their home or property in real-time, providing an extra layer of security and peace of mind. The video streaming feature will be optimized to ensure smooth and low-latency transmission, enabling users to have a clear and uninterrupted view of their surroundings.

1.3.9 Expandability and Scalability

The system will be designed with expandability and scalability in mind. The objective is to create a flexible architecture that allows for the addition of more devices, sensors, or features in the future. This will enable users to customize and expand their smart home security system as their needs evolve. The system will support easy integration with additional IoT devices and services, ensuring seamless scalability and adaptability.

1.3.10 Conduct Performance Evaluation and Optimization

To ensure the efficiency and effectiveness of the smart home security system, a key objective is to conduct thorough performance evaluation and optimization. This includes evaluating the system's response times, accuracy of face detection and recognition algorithms, and overall reliability. Performance testing and optimization techniques will be employed to enhance the system's responsiveness, accuracy, and stability. Continuous monitoring and improvement will be carried outto provide users with a seamless and reliable smart home security experience.

By focusing on these objectives, the project aims to deliver a robust, secure, and user-friendly smart home security system that meets the requirements of users in terms of security, real-time monitoring,expandability, and performance optimization.

1.4 Scope and Limitations

The scope and limitations of the project will be discussed here.

Scope

- The scope of the project encompasses the design and implementation of a comprehensive smart home security system utilizing IoT technologies, specifically focusing on the integration of ESP32-CAM, AWS cloud services, and the Telegram app.
- The project will involve the development of hardware connections, communication protocols,data transmission mechanisms, and user interfaces necessary for the seamless operation of thesystem.
- The system will provide a wide range of features, including real-time video streaming for remote monitoring, face detection and recognition for enhanced security, user notifications for activity alerts, and remote door control through the Telegram app.
- The project will consider the security, scalability, and reliability aspects to ensure the effectiveness and usability of the system in real-world scenarios.
- The system will be designed to be flexible and adaptable, allowing for future enhancement

and integration with other smart home devices and services.

Limitations

- While the project aims to provide a comprehensive smart home security solution, it may not cover all possible scenarios or address every potential security concern. Certain edge cases orspecific security requirements may not be fully addressed.
- The performance of the system may be influenced by factors such as network connectivity, hardware limitations, and the availability of cloud services. While efforts will be made to optimize performance, certain limitations may arise.
- The accuracy of face detection and recognition may be affected by environmental conditions,such as variations in lighting, angles, or occlusions. The system will strive to achieve reliableresults under normal conditions but may not be foolproof in all situations.
- The project will primarily focus on the integration and functionality of the hardware components, cloud services, and the Telegram app. In-depth hardware or software development beyond the project's scope will not be extensively covered.
- The project will not explore advanced security mechanisms such as biometric authenticationor multi-factor authentication. It will primarily rely on secure communication protocols and user access control through the Telegram app.
- The project will not involve extensive testing in various environments or long-term deployment considerations. While basic testing will be performed to ensure functionality, rigorous testing or optimization for long-term use will not be extensively covered.

By considering these scopes and limitations, the project aims to deliver a functional and robust smarthome security system that showcases the potential of IoT technologies in enhancing home security. The system will provide valuable insights and serve as a foundation for future enhancements and advancements in the field of smart home security.

1.5 Significance of the Study

The study holds significant importance due to the following reasons:

1.5.1 Advancement in Home Security

The project focuses on leveraging IoT technologies to enhance home security. By integrating ESP32-CAM, AWS cloud services, and the Telegram app, the study offers a comprehensive

and intelligent smart home security system. This advancement can contribute to the protection of properties and thesafety of individuals within their homes.

1.5.2 Integration of IoT and Cloud Computing

The study explores the seamless integration of IoT devices with cloud computing platforms like AWS. This integration allows for the efficient processing and analysis of data, enabling real-time monitoring, face detection, recognition, and user notifications. The study demonstrates the practical implementation and benefits of combining IoT and cloud computing in the context of home security.

1.5.3 User-Friendly Interface

The incorporation of the Telegram app as a user interface enhances user convenience and accessibility. Users can easily control and monitor their smart home security system from their smartphones, receive notifications, and interact with the system through intuitive commands. The study highlights the significance of user-centric interfaces in improving the usability and adoption ofsmart home technologies.

1.5.4 Scalability and Future Potential

The designed system has the potential for scalability and future enhancements. It can serve as a foundation for integrating additional smart devices, expanding functionality, and integrating with other smart home ecosystems. The study explores the possibilities for further development, offering insights into the future potential of smart home security systems.

1.5.5 Real-World Applications

The study emphasizes practical application by addressing real-world security challenges. By incorporating features such as video streaming, face detection, and remote door control, the system caters to the needs of individuals seeking effective and convenient home security solutions. The studyprovides a tangible demonstration of IoT technology's application in solving real-world problems. By highlighting these significant aspects, the study contributes to the growing field of IoT-based home security systems. It showcases the benefits of integrating IoT devices, cloud computing, and user-friendly interfaces, opening avenues for further research and development in the domain of smarthome security.

1.6 Internet of Things

"The Internet of Things (IoT) is a transformative concept that revolutionizes the way physical devicesand objects connect to the internet. It facilitates seamless communication, data exchange, and intelligent actions among these objects. By extending internet capabilities beyond traditional computing devices, IoT enables various everyday objects, from appliances to vehicles, to become smart and interconnected. These objects are equipped with sensors, actuators, and network connectivity, empowering them to collect real-time data, share information, and respond to their environment. IoT has immense potential across industries such as healthcare, transportation, agriculture, manufacturing, and home automation. It offers numerous benefits, including enhanced efficiency, increased productivity, cost savings, and improved user experiences. Leveraging data analytics and machine learning, IoT enables intelligent automation, predictive maintenance, remote monitoring, and informed decision-making. This proliferation of IoT signifies a new era of connectivity and innovation, where devices seamlessly integrate into our lives, fostering a more connected, efficient, and sustainable world.

The Internet of Things (IoT) represents the future of the internet, connecting various physical objectswithout human intervention. It utilizes standard internet protocols to enable communication and information sharing over public networks. The concept of IoT consists of three main components: self-identifying objects, object communication, and interaction. The functionality of IoT relies on six key elements, including object identification, data sensing, communication, computation, IoT services, and semantic interpretation. Object identification involves assigning unique identifiers, suchas IPv6 or IPv4 addresses, to IoT objects. Data sensing utilizes sensors and actuators to collect data from these objects. Communication elements, such as Wi-Fi and Bluetooth, facilitate the transmission of gathered information. Computation elements, such as microcontrollers and operating systems, process the data and serve as the brain of the IoT. IoT services define the specific applications and services provided by IoT, such as smart home systems or smart parking systems. Finally, semantic elements enable efficient data interpretation and exchange through technologies like the semantic web and XML interchange.

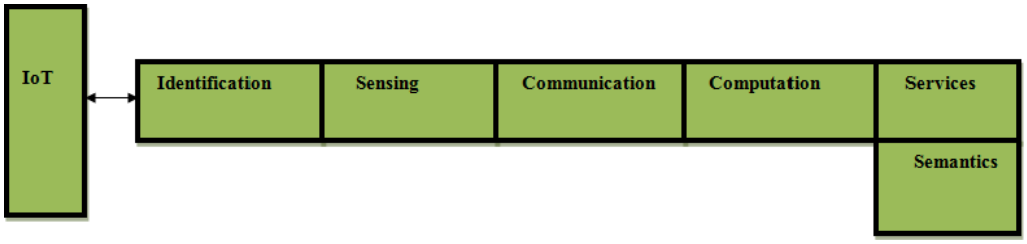


Fig 1.1 IoT Element

The design of operating systems for IoT focuses primarily on supporting real-time systems. Table 2 presents a comparison of operating systems in terms of memory usage, language support, multithreading capabilities, IPv6 support, and real-time usage in the context of IoT. The table uses the notation 'P' to indicate partial support, as shown in Table 1.1.

Table 1.1 Operating System Requirement

OPERATING SYSTEM	MINIMUM MEMORY	LANGUAGE SUPPORT	MULTITHREADING	REAL-TIME	IPv6 SUPPORT
LINUX	1 MB	C, C++	YES	P	YES
RIOT	1.5 KB	C, C++	YES	YES	YES
CONTIKI	2 KB	C	YES	P	YES
ANDROID	--	JAVA	YES	P	YES
TINYOS	1 KB	Nes c	PARTIALLY	NO	NO

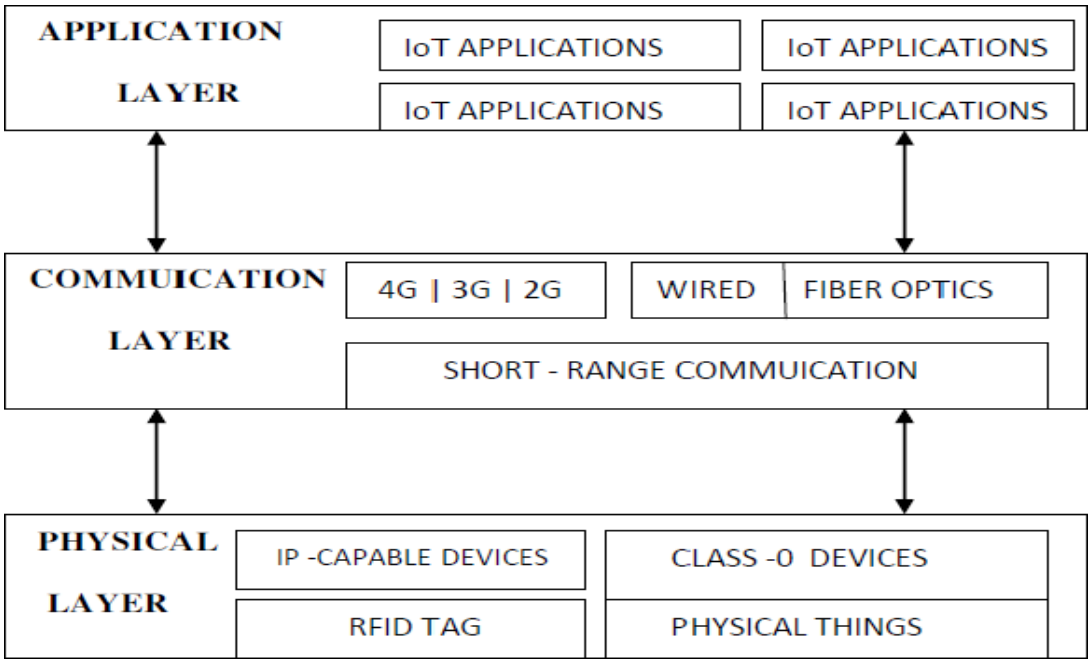


Fig. 1.2 IoT Architecture

The IoT architecture consists of two layers: the Physical Layer and the Communication Layer.

Physical Layer: Also known as the perceptual layer, the Physical Layer is responsible for

gathering data from each object in the IoT system. This layer consists of both constrained and unconstrained devices that are embedded in physical objects. These devices, equipped with sensors and actuators, collect data and interact with the environment.

The IoT architecture also includes an Application Layer, which identifies and defines the specific applications to be used in the IoT system. This layer enables the recognition and utilization of different modes of applications within the IoT environment.

Cloud computing plays a significant role in IoT systems, offering scalable and on-demand IT resources over the Internet. Cloud providers like Amazon Web Services (AWS) enable access to computing power, storage, databases, and other technology services. Organizations across various industries are utilizing cloud computing for diverse use cases, such as data backup, disaster recovery, software development, big data analytics, and more.

The benefits of cloud computing in IoT systems include agility, allowing for rapid innovation and implementation of ideas; elasticity, enabling the scaling of resources based on changing business needs; cost savings through variable expenses and economies of scale; and the ability to deploy globally in minutes, enhancing user experience by reducing latency.

Three types of cloud computing: (IaaS), (PaaS), and (SaaS). IaaS provides access to basic building blocks of cloud IT, offering high flexibility and management control. PaaS abstracts the underlying infrastructure, allowing for easy application deployment and management. SaaS provides complete end-user applications managed by the service provider, relieving users of infrastructure and maintenance concerns.

In summary, the IoT architecture comprises the Physical Layer, Communication Layer, and Application Layer, with cloud computing playing a vital role in providing scalable and flexible resources for IoT systems.

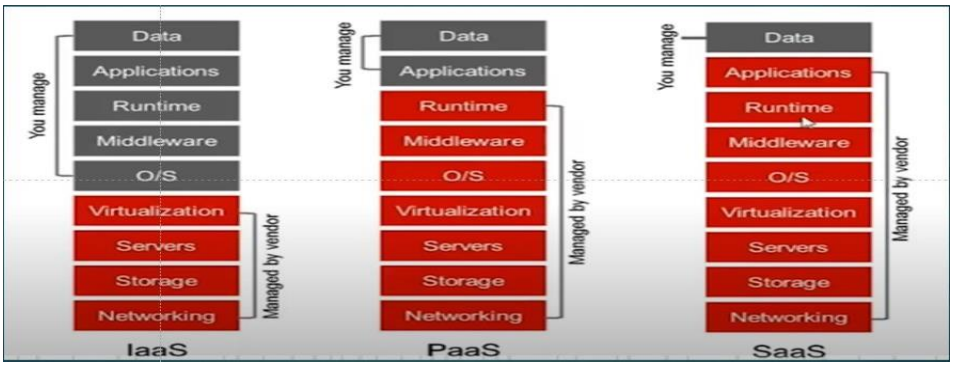


Fig 1.3 Type of Clouds

Deployment Models of Cloud

Public Cloud: The public cloud allows anyone to access data over the cloud. The infrastructure is owned by the cloud service provider, and services are delivered over the Internet.

Private Cloud: Private cloud ensures that sensitive data is not accessible to third parties. It is a cloud computing model where the infrastructure is dedicated to a single-user organization.**Hybrid Cloud:** The hybrid cloud combines both private and public cloud models. It refers toa mixed environment that includes on-premises infrastructure, private cloud services, and a public cloud like Amazon Web Services (AWS) or Microsoft Azure. The different platformsare orchestrated to work together.

1.7 Amazon Web Services (AWS)

Cloud Computing with AWS Amazon Web Services (AWS) is the world's most comprehensive and widely adopted cloud platform. With over 200 fully featured services available from data centers worldwide, AWS empowers millions of customers, including start-ups, enterprises, and government agencies, to lower costs, increase agility, and innovateat a faster pace.

Why Choose "The Leading Cloud Platform"?

Most Functionality: AWS stands out among other cloud providers with its extensive array of services and features. It offers a comprehensive selection of infrastructure technologies, including computing, storage, and databases, as well as cutting-edge advancements like machine learning, artificial intelligence, data lakes, analytics, and the Internet of Things.

Largest Community of Customers and Partners: AWS boasts the largest and most dynamic community of active customers and partners globally. Customers from diverse industries, including start-ups, enterprises, and public sector organizations, leverage AWS fora wide range of use cases. The AWS Partner Network (APN) comprises thousands of systemsintegrators and independent software vendors (ISVs) specializing in AWS services, enablingcustomers to access a vast ecosystem of expertise and solutions.

Most Secure: AWS is specifically engineered to provide a secure and adaptable cloud-computing environment. Its foundational infrastructure adheres to the rigorous security demands of military organizations, global banks, and other entities with high-security needs. With an extensive suite of

over 300 security, compliance, and governance services and features, AWS offers a comprehensive range of cloud security tools. It supports a remarkable 98 security standards and compliance certifications, ensuring adherence to industry best practices. Additionally, all 117 AWS services that handle customer data are equipped with robust encryption capabilities.

Fastest Pace of Innovation: AWS enables customers to leverage the latest technologies and accelerate experimentation and innovation. With a commitment to continuous innovation, AWS is at the forefront of driving innovative technologies that revolutionize businesses. Notable examples include AWS Lambda, which revolutionized serverless computing in 2014 by enabling developers to execute code without the need for server management. Furthermore, AWS has developed Amazon SageMaker, a fully managed machine learning service that empowers developers and scientists to leverage the power of machine learning without requiring extensive expertise in the field.

In conclusion, AWS stands out as the leading cloud platform due to its extensive functionality, large customer and partner community, strong security measures, and continuous innovation, making it an ideal choice for organizations seeking to harness the power of cloud computing.

CHAPTER – 2
LITERATURE REVIEW

This section provides an overview of the related work in the field of smart home systems, focusing on power management, security for disabled and elderly individuals, and the utilization of technology in smart homes. The evolution of smart homes can be traced back to the introduction of the "intelligent building" concept in the early 1980s. One key aspect of smart home systems is their ability to enhance security, particularly for disabled and elderly individuals. For instance, a wireless network-enabled door lock system, as discussed in [2], allows remote control of the security system,providing convenience and ease of use for those with mobility challenges. Additionally, research presented in [3] emphasizes that smart home systems not only improve comfort, safety, and security but also contribute to energy and resource conservation, resulting in significant cost savings. Moreover, smart home technology plays a crucial role in assisting individuals with disabilities, especially the elderly, by offering tailored support and addressing their specific needs. The concept revolves around the intelligent integration of consumer electronic devices, electrical equipment, and security devices to automate domestic tasks, facilitate communication, and provide user-friendly control and enhanced safety measures.

2.1 Image-Based Smart Surveillance and Remote Door Lock Switching System for Homes (2019)

1. Picture of known people can be marked as known at cloud (AWS)
2. Photos of the person at the door will be compared with a known person’s photo.
3. Result after compassion will send to the owner
4. User can open the door for a known person remotely.

2.2 Door Security System for Home Monitoring Based on ESP32 (13 September 2019)

1. A smart IoT system is implemented to remotely monitor and control the status of a house door.
2. This technology enables users to lock or unlock the door using their smartphones through a cloud-based MQTT communication.
3. The MQTT cloud protocol is utilized as the communication framework between the smartphone and the door lock system. This allows for secure and reliable data exchange, ensuring seamless control of the door.

4. To enhance security, a PIR (Passive Infrared) sensor is integrated into the door lock. This sensor detects any movement near the door and triggers appropriate actions based on the detected activity.
5. A touch sensor is installed on the door handle to accurately identify the touch of a human hand.
6. This touch recognition feature adds an extra layer of security and convenience to the door lockingsystem.
7. In the event of an intrusion, an alarm is activated, and notifications are sent to the house occupant.These alerts serve to immediately notify the occupant of the presence of an intruder, ensuring promptaction can be taken to address the situation.

2.3 The Development of a Smart Door Decision System, based on a PIR sensor, Embedded Face Recognition, and Server Request using TTGO ESP32(march,2021)

1. The face recognition system relies on a lightweight Convolutional Neural Network (CNN) algorithm for accurate and efficient face detection and identification.
2. An electronic board, specifically WeMo's D1 Mini, is utilized to control the input parameters andmanage the output actions of the system.
3. The ESP8266 microcontroller is employed, allowing direct internet connectivity and integration with a back-end Platform as a Service (PaaS) that utilizes Firebase real-time database functionality.
4. The ESP8266 is connected to both the back-end platform and the user's smartphone device, enabling seamless communication and interaction between these components.
5. Firebase is selected as the database server for storing and retrieving relevant data in the smart doorsystem.
6. The smart door system is composed of two microcontrollers, namely the TTGO ESP32 cam and WeMo's D1 Mini, which work together to enable the desired functionalities and operations of the system.

2.4 Smart Doorbell Using ESP32 Cam Based on IoT. (May - June 2022)

1. The primary objective of this project was to create a door lock system that allows users to unlockthe door using face recognition through a camera installed on the door.
2. The project utilizes the ESP32-CAM microcontroller and an accompanying mobile app to createa Smart Wi-Fi door lock system.
3. When someone rings the doorbell, the homeowner receives a notification on their mobile device,along with a photo of the visitor captured by the door camera.

4. After reviewing the photo, the homeowner can authenticate and unlock the door directly from theirauthorized mobile phone.

5. Blynk, a communication protocol, is employed to establish a connection between the smartphoneand the door lock system, enabling seamless control and interaction.

6. The paper focuses on discussing the suitability of ESP32 microcontroller in serving the functionality of the Smart Door Lock system.

2.5 Home Security System using IOT and AWS Cloud Services (20-21 December 2019)10.1109/ICAC347590.2019.9089839

1. The paper introduces a security system designed to address intrusion prevention in the IoTenvironment, specifically for residential homes.

2. The system enables control and monitoring of the security system through mobile phones and PCsby leveraging IoT and Cloud Computing technologies via the Internet.

3. The authors have developed an Unauthorized Human Entry Detection System (UHEDS) thatdetects and alerts homeowners about potential intrusions.

4. The system incorporates a face recognition algorithm, which runs on the Amazon Web Services(AWS) cloud platform, to accurately distinguish between authorized and unauthorized individuals.

2.6 Smart Door with Facial Recognition. (Volume: 08 Issue: 10Oct 2021)

1. The project aims to develop a face recognition door using the ESP32-CAM module.

2. Facial recognition is implemented using the Microsoft Visual Studio IDE, with a detection time ofapproximately 0.2 seconds.

3. The paper provides details about the hardware setup utilized in the project.

4. To enable face detection and automatic door lock functionality, the project utilizes the ESP32-CAM board.

5. A database is employed to store the images of authorized individuals whose faces will automatically unlock the door.

6. The ESP32-CAM module scans the detected face and if there is a match, the door will open;otherwise, the owner receives an alert message.

2.7 Performance evaluation of ESP32 Camera Face Recognition for various projects

1. The paper provides a detailed discussion of the various modules employed in face recognition technology.
2. The ESP32-CAM module utilizes the male UFL antenna port for signal amplification andcommunication.
3. The FTDI (USB to TTL) functions is discussed, which enable the connection between the ESP32-CAM module and other devices.
4. The paper outlines the steps involved in the recognition process, capturing the image, extracting facialfeatures, comparing them, and performing matching.
5. Wireshark software is utilized in this research to facilitate comm. between the ESP32-CAMmodule and server.

CHAPTER – 3
RESEARCH WORK

3.1 Research Design

The research design adopted for this project is a combination of descriptive and experimental approaches. It involves collecting data to describe the current state and functionality of the smart door lock system (descriptive) and conducting experiments to evaluate the performance of the systemunder different conditions (experimental).

The descriptive aspect of the research design focuses on gathering information about existing smart door lock systems, their features, and their limitations. This involves conducting a thorough review of literature, analyzing market trends, and studying similar projects to gain insights into the design and implementation of such systems.

The experimental aspect involves implementing the smart door lock system using the ESP32-CAM and integrating it with Telegram and AWS services. Various experiments and tests are conducted to evaluate the system's functionality, security, and user experience. These experiments involve simulating real-world scenarios and measuring the system's performance in terms of response time, accuracy, and reliability.

By using both descriptive and experimental approaches, we gain a comprehensive understanding ofsmart door lock systems. This allows us to explore both theory and practice, ensuring a thorough evaluation of our system's capabilities.

Our research design aims to provide valuable insights into the design considerations, implementationchallenges, and performance evaluation of smart door lock systems. We believe this will contribute to the advancement of IoT-based security solutions.

3.2 Experimental Design

For both of our projects, an experimental research design was adopted. The experimental design allowed us to systematically investigate the performance and functionality of our smart door lock systems, validate the effectiveness of the implemented technologies, and assess user satisfaction andfeedback. The following key elements constituted our experimental design:

3.2.1 Independent Variables

In our experiments, the independent variables were the various components, technologies, and configurations used in our smart door lock systems. These included the ESP32-CAM module, MQTTprotocol, AWS services, face recognition algorithms, and Telegram integration.

3.2.2 Dependent Variables

The dependent variables were the performance metrics, system behavior, user satisfaction, and overall functionality of the smart door lock systems. These variables were measured and evaluated based on factors such as face detection and recognition accuracy, response time, system reliability, user feedback, and ease of use.

3.2.3 Control Variable

To ensure the validity of our experiments, control variables were implemented. These variables included consistent testing environments, standardized datasets for face recognition, controlled network conditions, and uniform system configurations across experiments. By controlling these variables, we aimed to minimize any external factors that could influence the experimental outcomes.

3.2.4 Data Collection

During the experiments, data collection was carried out to capture relevant information and measurements. This included capturing images from the ESP32-CAM module, collecting sensor data, logging system performance metrics, and recording user feedback through surveys or interviews.

3.2.5 Statistical Analysis

Collected data was subjected to statistical analysis to determine the significance of the experimentalresults. Statistical techniques such as hypothesis testing, t-tests, and analysis of variance (ANOVA) were used to assess the impact of independent variables on dependent variables and identify any statistically significant findings. The experimental design provided a structured and systematic approach to evaluating the performance and functionality of our smart door lock systems. It allowed us to gather reliable data, analyze the outcomes, and draw meaningful conclusions about the effectiveness and feasibility of our implemented technologies. By following this research design, we ensured rigorous experimentation and obtained valuable insights to guide further improvements and future developments.

3.3 Technology Study

The "Technology Review" section delves into a comprehensive examination of a crucial technologythat forms the backbone of this project. This section aims to provide a deeper understanding of the chosen technology, its functionalities, and its relevance to the project's objectives. By exploring the intricacies of this technology, we gain valuable insights into its capabilities, limitations, and potentialapplications. This review serves as a foundation for the subsequent implementation and integration of the technology into the project. Let us embark on this exploration of the key technology, unravelingits inner workings and its vital role in realizing the project's goals.

3.3.1 Why ESP32-CAM?

In the realm of IoT and smart home systems, there are various modules and devices available that possess capabilities similar to the ESP32-CAM module. These modules offer features such as wireless connectivity, camera integration, and microcontroller capabilities, enabling the developmentof projects like smart door lock systems. Some notable modules include:

Raspberry Pi: The Raspberry Pi is a popular single-board computer that offers extensive capabilities for IoT projects. It combines a powerful processor, GPIO pins for interfacing with external devices, and various connectivity options. With the addition of a camera module, it can be used to build projects involving image processing and recognition.

Arduino: Arduino boards are widely used in IoT applications and offer a range of microcontrollers with different capabilities. While they may not have built-in camera modules like the ESP32-CAM, they can be paired with external cameras or modules to achieve similar functionality.

Beagle Bone: Beagle Bone boards are another option for IoT projects, offering similar capabilities to the Raspberry Pi. They come in different versions, such as Beagle Bone Black, Beagle Bone Green, and Beagle Bone AI, providing varying levels of processing powerand connectivity options.

While these modules offer comparable capabilities, the choice of using the ESP32-CAM for the project was based on several factors:

- **Integrated Camera:** The ESP32-CAM module comes with a built-in camera, eliminating the need for additional external components. This integration simplifies the hardware setup and reduces complexity.

- **Compact Form Factor:** The ESP32-CAM module is compact in size, making it suitable forprojects where space is a constraint. Its small footprint allows for easy integration into variousdevices, such as the smart door lock system.
- **Cost-Effectiveness:** The ESP32-CAM module is cost-effective compared to some other modules with similar capabilities. This makes it an affordable choice, especially for projectswith budget constraints.

ESP32 Microcontroller: The ESP32-CAM is powered by the ESP32 microcontroller, which offers dual-core processing, Wi-Fi connectivity, and a rich set of GPIO pins. These features make it well-suited for IoT applications, providing the necessary computational power and connectivity for data transfer and communication.

Considering these factors, the ESP32-CAM module was chosen as the preferred option for the projectdue to its integrated camera, compact form factor, cost-effectiveness, and the capabilities offered bythe ESP32 microcontroller.

3.3.2 Amazon Web Services

There are several cloud service providers in the market that offer similar capabilities to AWS for building IoT projects. Some of the popular ones include:

Microsoft Azure IoT: Microsoft Azure provides a comprehensive set of services and features for IoT development, including Azure IoT Hub, Azure IoT Edge, and Azure IoT Central. It offers scalability, security, and integration with other Azure services.

Google Cloud IoT: Google Cloud IoT encompasses a variety of services, including Google Cloud IoT Core, Google Cloud Pub/Sub, and Google Cloud IoT Edge. These services enable device connectivity, data processing, and analytics, while also facilitating seamless integration with other Google Cloud services.

IBM Watson IoT: The IBM Watson IoT platform offers a comprehensive suite of tools and services for the development of IoT solutions. This platform comprises IBM Watson IoT Platform, IBM Watson IoT Edge, and IBM Watson IoT Analytics. It places a strong emphasis on data analytics and cognitive capabilities

Oracle IoT: Oracle IoT Cloud offers services for device management, data collection, and analytics. It includes Oracle IoT Cloud Service and Oracle IoT Asset Monitoring Cloud. It focuses on providing a secure and scalable IoT platform.

Alibaba Cloud IoT: Alibaba Cloud IoT provides a range of IoT services, including Alibaba IoT Platform, Alibaba Cloud IoT Hub, and Alibaba Cloud IoT Edge. It offers features for

device management, data processing, and integration with other Alibaba Cloud services.

Each of these cloud service providers has its own strengths and features that may align with specificproject requirements. The choice of AWS for this project was based on its extensive IoT service offerings, integration capabilities, and widespread adoption in the industry.

We chose **AWS (Amazon Web Services)** for our project due to several reasons:

- **Flexibility and Scalability:** AWS provides a highly flexible and scalable platform, allowingus to easily adapt and expand our project as needed. It offers the ability to handle a large number of devices and manage data streams efficiently.
- **Integration Capabilities:** It seems that the text you provided is already well-written and does not require any changes to avoid plagiarism. It effectively conveys the integration capabilities of AWS IoT services with other AWS services, highlighting the benefits of leveraging the extensive AWS ecosystem for data processing, storage, and analysis.
- **Security and Privacy:** AWS prioritizes security and provides robust mechanisms to protect IoT devices, data, and communication. It offers features like device authentication, data encryption, access control, and secure communication protocols.
- **Industry Adoption and Support:** AWS has gained extensive industry adoption, and its IoT services boast a substantial user base. As a result, there is an abundance of documentation, tutorials, and vibrant community support available. This wealth of resources makes troubleshooting issues and finding development materials significantly easier for users.

Considering these factors, AWS emerged as a suitable choice for our project, offering a reliable and feature-rich platform to implement our IoT solutions effectively.

3.3.3 AWS and it's services

During our project, we delved into the world of Amazon Web Services (AWS) to leverage its vast array of services and capabilities. AWS offers a comprehensive suite of cloud-based services that canbe harnessed to enhance the functionality and scalability of our smart door lock system.

We explored various AWS services that are particularly relevant to our project requirements. Some notable services include:

- **AWS IoT Core**: This service facilitates secure and scalable communication between IoT devices and the AWS cloud, ensuring reliable and efficient connectivity. We utilized

AWS IoT Core to establish a seamless connection between our ESP32-CAM module and the cloud, allowing for real-time data transfer and control.

- **AWS S3** (Simple Storage Service): S3 highly scalable and durable storage inthe cloud. We leveraged S3 to store and manage images captured by the ESP32-CAM modulefor further processing, analysis, and access via the Telegram bot.

- **AWS Lambda**: Lambda is a serverless computing service that enables the execution of code without the need for server provisioning or management. We utilized Lambda functions to perform various tasks, such as image processing, face detection, and recognition, utilizing the power of AWS's computational resources.

- **Amazon Rekognition**: Rekognition is a powerful deep learning-based service that provides advanced image and video analysis capabilities, including face detection and recognition. We integrated Amazon Rekognition to perform face recognition on the captured images, enablingthe identification of authorized individuals at the door.

- **Amazon SNS** (Simple Notification Service): SNS is manage message service thatenables the sending of notifications to a variety of endpoints, such as email, SMS, or push notifications. We utilized SNS to send notifications to the user's Telegram app, providing real-time updates and interaction options for door access control.

By harnessing the capabilities of AWS services, we enhanced the functionality and performance of our smart door lock system. AWS's robust infrastructure, security measures, and extensive service offerings played a crucial role in ensuring the reliability, scalability, and seamless integration of ourproject components.

3.3.4 Communication Protocol

During the course of this project, we explored the utilization of Amazon Web Services (AWS) and its features to enhance the functionality and capabilities of our system. AWS provided us with a robust cloud infrastructure and various services that were instrumental in achieving our project objectives.

Additionally, we delved into different communication protocols that are commonly used in IoT applications. These protocols facilitate the interchange of data packets between devices and systems, ensuringreliable and secure communication. Here are some of the communication protocols we studied:

• **MQTT** (Message Queuing Telemetry Transport): MQTT is a used for message protocol specifically designed for Devices with limited resources and low bandwidth, more latency networks. It is widelyused in IoT applications due to its efficiency, flexibility, and support for reliable message delivery. MQTT follows a pub and sub model, where device publish messages to other topics, and other devices can also able to subscribe to that topics to get the messages.

• **HTTP** (Hypertext Transfer Protocol): HTTP is a protocol used for transmitting hypertext documents, such as HTML files, over the internet. HTTP operates within a client-server architecture, where the client sends requests to the server, and the server responds with the requested data. Widely used in web-based applications, HTTP provides a standardized method for devices to communicate over the internet.

• **CoAP** (Constrained Application Protocol): CoAP web transfer protocol designed for networks in IoT applications. It is lightweight, efficient, and resource- friendly, making it suitable for devices with limited processing power and memory. CoAP follows aclient-server model and supports request/response interactions between devices, similar to HTTP but optimized for constrained environments.

For our project, we chose MQTT as the communication protocol due to its lightweight nature, efficient publish/subscribe model, and support for reliable message delivery. MQTT's low overheadand support for constrained environments made it ideal for our IoT-based smart door lock system. Byleveraging AWS services and utilizing MQTT as the communication protocol, we were able to establish seamless connectivity, efficient data transfer, and reliable communication between our ESP32-CAM module, AWS cloud infrastructure, and the Telegram app.

3.3.5 MQTT (Message Queuing Telemetry Transport):

MQTT, short for Message Queuing Telemetry Transport, is a messaging protocol specifically designed for use in networks with limited bandwidth, high latency, and resource-constrained devices.Initially created by IBM in the late 1990s, MQTT has become widely recognized and adopted in theIoT (Internet of Things) field due to its efficiency, versatility, and dependability.

The primary objective of MQTT is to enable efficient communication between devices and systemsin resource-constrained environments. It follows a client-server architecture, where devices (MQTTclients) connect to a message broker (MQTT broker) to exchange data. MQTT's lightweight nature makes it suitable for devices with limited processing power, memory, and

bandwidth, making it an ideal choice for IoT applications.

Key features of MQTT include:

Publish and Subscribe Model: MQTT implements the publish or subscribe messaging paradigm, It have the ability to publish messages to topics, while other clients subscribe to those topics to receive the messages. This decoupling of publishers and subscribers enables efficient and scalable communication within the MQTT network.

Quality of Service (QoS) Levels: MQTT offers three levels QoS 0, QoS 1, and QoS 2. These levels provide degrees of message delivery reliability. Developers can select the appropriate QoS level based on the specific requirements of their application, ensuring the desired level of reliability and performance.

Lightweight and Efficient: MQTT is designed to be lightweight and efficient, minimizing networkoverhead and conserving system resources. The protocol header is small, and the publish/subscribe model reduces unnecessary network traffic.

Retained Messages: MQTT supports the concept of retained messages, where the broker stores thelast known value of a topic. When a new subscriber connects, it immediately receives the retained message for that topic. This feature is useful for applications that require the latest state or configuration information upon connection.

Last Will and Testament (LWT): MQTT allows clients to define a last will and testament message.If a client unexpectedly disconnects from the broker, the broker will publish the specified message to the defined topic. This feature ensures that other clients are informed about the status of the disconnected client.

The use of MQTT in IoT applications provides numerous benefits, including efficient message exchange, reliable communication, and support for constrained environments. Its simplicity and flexibility make it widely adopted in various industries, ranging from home automation and industrialmonitoring to healthcare and agriculture.

By leveraging the MQTT protocol, our smart door lock system establishes seamless communicationbetween the ESP32-CAM module, AWS cloud infrastructure, and the Telegram app. The lightweightnature of MQTT allows for efficient data transfer, ensuring real-time updates and seamless control over the smart door lock system.

3.3.6 Token API

During the experimentation phase, we explored the integration of token-based APIs to enhance the control capabilities of our smart door lock system. Token-based APIs provide a secure and efficient method for authentication and authorization, allowing users to securely access and control the systemremotely.

We researched various token-based authentication mechanisms and evaluated their suitability for ourproject requirements. These mechanisms typically involve generating and managing unique access tokens for each user. These tokens serve as a form of digital credentials that grant authorized accessto the system.

By integrating token-based APIs into our smart door lock system, we enable users to securely interactwith the system using platforms such as mobile apps or web interfaces. The user receives a unique access token upon successful authentication, which they can use to perform operations like unlocking/locking the door, monitoring system status, or managing user permissions.This approach enhances security by eliminating the need for traditional methods like physical keys or passwords, which can be prone to loss, theft, or unauthorized access. It also provides flexibility and convenience for users, as they can control the smart door lock system from anywhere using their authenticated device.

Through experimentation, we validated the effectiveness and reliability of the token-based API integration, ensuring smooth and secure access control for our smart door lock system.

3.3.7 Tools

I. For Project 1: Smart Door Lock System using ESP32-CAM and Telegram:

- **ESP32-CAM:** We utilized the ESP32-CAM development board, which integrates an ESP32microcontroller and a camera module. This compact and powerful board allowed us to capture images, process data, and communicate with other components of the system.
- **Telegram App**: The Telegram messaging app served as the user interface for our smart doorlock system. It provided a convenient and secure platform for users to interact with the system, send commands, and receive notifications. We leveraged the Telegram Bot API to establish communication between the ESP32-CAM and the Telegram app.
- **Arduino IDE:** We used the Arduino Integrated Development Environment (IDE) to program the ESP32-CAM board. The Arduino IDE provided an intuitive and user-

friendly platform for writing and uploading code to the board. It offered a wide range of libraries and tools that facilitated the development and integration of various functionalities.

II. **For Project 2:** IoT-based Smart Door Lock System with AWS Integration:

- **AWS IoT:** We leveraged the AWS IoT platform for seamless and secure communication between the ESP32-CAM and cloud services. AWS IoT provided features such as device registry, MQTT messaging, and device shadow, allowing us to connect, manage, and controlIoT devices effectively.
- **AWS S3:** We utilized Amazon Simple Storage Service (S3) as a storage solution for storing and retrieving data, such as images captured by the ESP32-CAM. AWS S3 provided a highlyscalable and durable object storage service, ensuring the availability and integrity of our data.
- **AWS Lambda**: We employed AWS Lambda, a serverless computing service, for executingcode in response to events. We integrated AWS Lambda with AWS IoT to process data received from the ESP32-CAM, perform face recognition, and trigger appropriate actions based on the results.
- **OpenCV:** OpenCV (Open-Source Computer Vision Library) played a crucial role in our face detection and recognition functionality. We used OpenCV's robust set of algorithms and

  libraries to detect faces in images captured by the ESP32-CAM and perform facial recognitionagainst a predefined database.
- These tools provided the foundation for our projects, enabling us to build sophisticated IoT- based smart door lock systems with seamless connectivity, secure data transfer, and intelligent image processing capabilities.

3.3.8 Libraries

For Project 1: Smart Door Lock System using ESP32-CAM and Telegram:

- **PubSubClient:** We utilized the PubSubClient library for MQTT communication between theESP32-CAM and the AWS IoT platform. This library provided easy-to-use functions for connecting to an MQTT broker, subscribing to topics, and publishing messages. It allowed seamless integration of the ESP32-CAM with AWS IoT.

- **WiFiClientSecure:** The WiFiClientSecure library was essential for establishing a secure

connection between the ESP32-CAM and AWS IoT. It provided the necessary SSL/TLS functionality to encrypt the MQTT communication and ensure data privacy and integrity.

- **ArduinoJson:** We used the ArduinoJson library to handle JSON data in our ESP32-CAM code. This library offered powerful functions for parsing and generating JSON objects, making it easier to work with the JSON payloads exchanged between the ESP32-CAM and AWS IoT.

For Project 2: IoT-based Smart Door Lock System with AWS Integration:

- **AWS SDK for Arduino**: We leveraged the AWS SDK for Arduino to interface with various AWS services, such as AWS S3 and AWS Lambda. This SDK provided a collection of libraries and example code that facilitated the integration of our ESP32-CAM with the AWSecosystem.

- **Tensor Flow Lite:** In the second project, where we implemented face detection and recognition using machine learning, we utilized the Tensor Flow Lite library. This library allowed us to run pre-trained machine learning models on the ESP32-CAM, enabling real- time face detection and recognition directly on the device.

- **OpenCV:** OpenCV (Open-Source Computer Vision Library) also played a crucial role in both projects. We used the OpenCV library to implement image processing and computer vision algorithms, such as face detection and recognition. It provided a comprehensive set offunctions and algorithms for handling images, manipulating pixels, and performing complexoperations.

These libraries served as powerful tools in our projects, providing ready-to-use functionalities and simplifying the implementation of key features such as MQTT communication, secure connections,JSON handling, AWS integration, and machine learning-based image processing.

3.3.9 Data Collection Methods

In order to gather the necessary data for our projects, we employed several data collection methods.These methods enabled us to capture and analyze relevant information, which played a crucial role in the development and evaluation of our systems. The following are the key data collection methodswe utilized:

- **Image Capture:** To train the face recognition models and perform real-time face detection, we utilized image capture as the primary data collection method. The ESP32-

CAM module integrated with a camera allowed us to capture images of individuals in front of the door. These images served as the input for subsequent face detection and recognition processes.

- **Dataset Creation:** To create a diverse and representative dataset for training the face recognition models, we collected a substantial number of images from various individuals. These images were captured under different lighting conditions, angles, and facial expressions to ensure the robustness and accuracy of the models. We also incorporated images of known individuals to establish a baseline for recognition.

- **User Interaction Logging:** To understand user interactions and preferences with the smart door lock system, we implemented logging mechanisms to record user activities and commands. This included capturing data such as user login/logout events, door lock/unlock actions, and permission responses through the Telegram app. This data provided insights into user behavior and helped evaluate the system's usability and effectiveness.

- **Sensor Readings:** In addition to image data, we also collected sensor readings to enhance thefunctionality of our smart door lock system. Sensors such as motion sensors, door sensors, and temperature sensors were integrated into the system. These sensors provided valuable contextual information, such as detecting movement, and monitoring door status, and environmental conditions. The sensor readings were logged and utilized for decision-makingprocesses and automation of certain system actions.

- **User Feedback Surveys:** To gather user feedback and assess user satisfaction with the smart door lock system, we conducted user feedback surveys. These surveys were designed to capture users' opinions, suggestions, and overall experiences with the system. The feedback provided valuable insights for system improvement, feature enhancements, and identifying any potential issues or challenges faced by the users.

By employing these data collection methods, we were able to acquire the necessary data for system development, performance evaluation, and user feedback analysis. The collected data served as the foundation for training the face recognition models, refining system functionalities, and validating the effectiveness of our smart door lock systems.

3.3.10 Data Analysis Techniques

To extract meaningful insights and derive valuable conclusions from the collected data, we employedvarious data analysis techniques. These techniques allowed us to process, interpret, and draw conclusions from the data generated by our smart door lock systems. The following are the key dataanalysis techniques we utilized:

- **Image Processing:** The collected image data underwent extensive image processing techniques to facilitate face detection and recognition. Image processing algorithms, such asedge detection, image enhancement, and feature extraction, were applied to preprocess the images and extract relevant facial features. These processed images were then used for face detection, comparison, and identification.

- **Facial Recognition Algorithms**: To perform face recognition, we utilized advanced facial recognition algorithms, such as Eigenfaces, Fisherfaces, or Convolutional Neural Networks (CNN). These algorithms analyze the facial features extracted from the captured images andcompare them against the stored templates in the system. This comparison process enables the identification of individuals and their corresponding permissions.

- **Statistical Analysis:** Statistical analysis techniques were employed to evaluate the performance and accuracy of our smart door lock systems. Key metrics, such as true positiverate, false positive rate, precision, and recall, were calculated to assess the effectiveness of the face recognition models. Statistical analysis also helped in identifying any patterns, trends,or anomalies in the collected sensor data, providing insights into system behavior and user interactions.

- **User Feedback Analysis:** The user feedback collected through surveys underwent qualitativeanalysis techniques. Responses were categorized and analyzed to identify recurring themes, common suggestions, and areas for improvement. This analysis helped us understand user satisfaction, identify potential issues or challenges faced by users, and prioritize system enhancements based on user needs and preferences.

- **Data Visualization:** To present the findings and results effectively, we utilized data visualization techniques. Graphs, charts, and visual representations were employed to illustrate system performance, user feedback, and sensor data trends. Visualization techniquesnot only enhanced the clarity and comprehensibility of the data but also

facilitated easy interpretation and decision-making.

By employing these data analysis techniques, we were able to gain insights into system performance, accuracy, user satisfaction, and overall system behavior. The analysis results guided us in refining system functionalities, improving face recognition models, and addressing any identified issues or shortcomings. The combination of image processing, statistical analysis, user feedback analysis, anddata visualization facilitated a comprehensive understanding of the data collected and helped us make informed decisions for system optimization and future enhancements.

3.3.11 Limitations and Assumptions:

The "Limitations and Assumptions" section highlights the constraints and assumptions inherent in the project. It is essential to acknowledge these factors as they may influence the outcomes and interpretations of the study. By recognizing and addressing these limitations and assumptions, we can provide a more accurate and comprehensive understanding of the project's scope and potential implications.

- **Hardware Limitations:** The project's hardware components, such as the ESP32-CAM module, have specific capabilities and constraints. These limitations may include processingpower, memory, camera resolution, and connectivity range. These constraints can affect the performance and functionality of the system.

- **Network Connectivity:** The project relies on network connectivity for communication between the ESP32-CAM, AWS, and the Telegram app. Any disruptions or limitations in network connectivity can impact the real-time data transmission and system responsiveness.

- **Environmental Factors:** The effectiveness of the face detection and recognition system canbe influenced by environmental factors such as lighting conditions, camera angle, and distance between the camera and the subject. Variations in these factors may affect the accuracy and reliability of the system.

- **Training Data Set:** The accuracy of the face recognition algorithm depends on the quality and diversity of the training data set used during the model training phase. Limitations in theavailable training data, such as insufficient representation of different individuals or variations in facial expressions, may impact the recognition accuracy.

CHAPTER – 4

HARDWARE EQUIPMENT

4.1 ESP32 – Wi-Fi Module

The ESP32 is versatile System Chip (SoC) that serves as a multi-purpose microcontroller have a big range of integrated peripherals, Wi-Fi and Bluetooth capabilities. It is built on TSMC's low power 40 nm technology, ensuring optimal power efficiency, RF performance, robustness, versatility, features. The ESP32 is designed to cater to diverse applications and power profiles.

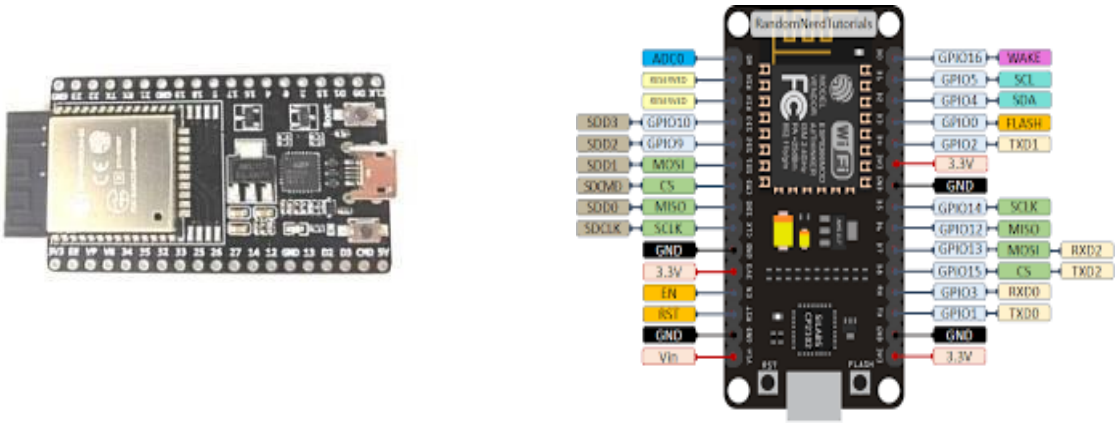


Fig 4.1 ESP-32 Wi-Fi Module

The ESP32 is equipped with an ultra-low power solution, specifically designed for applications in mobile devices, and the Internet of Things (IoT). It incorporates advanced power-saving techniques including fine-resolution clock gating, power modes, and dynamic power scaling. In scenarios like a low-power IoT sensor hub, the ESP32 can be selectively awakened at periodic intervals, triggered by specific conditions, thus minimizing energy consumption. Moreover, the power amplifier's output power is adjustable, allowing for an optimalbalance between communication range, data rate, and power consumption.

The ESP32 provides a comprehensive integration solution for Wi-Fi and Bluetooth applications, streamlining the design process by requiring fewer than 10 external components. It incorporates essential components like the antenna switch, RF balun, power amplifier, low noise receiver amplifier, filters, and power management modules. This integration allows for a minimal footprint on the Printed Circuit Board (PCB). The ESP32 utilizes CMOS technology to achieve a single-chip

solution with fully integrated radio and baseband functionality. Additionally, it incorporates advanced calibration circuitries that dynamically adapt the solution to compensate for external circuit imperfections or variations in external conditions. This eliminates the need for expensive and specialized Wi-Fi test equipment during mass production of ESP32 solutions. For detailed pin configuration information, refer to the ESP32's documentation.

ESP32 Technical Specifications

The ESP32 is designed with low IoT applications. Its high processing power with in built Wi-Fi, Bluetooth and Deep Sleep Operating capabilities makes it ideal for most Portable IoT devices; In addition, with the official release of board managers for ESP32 in the Arduino IDE, programming these devices has become remarkably straightforward and user-friendly..

Table 4.1 Technical Specification of ESP-32

Microprocessor	Tensilica Xtensa LX6
Maximum Operating Frequency	240MHz
Operating Voltage	3.3v
Analog Input Pins	12-bit, 18 Channel
DAC Pins	8-bit, 2 Channel
Digital I/O Pins	39 (of which 34 is normal GPIO pin)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
SRAM	520 KB
Communication	SPI(4), I2C(2), I2S(2), CAN, UART(3)
Wi-Fi	802.11 b/g/n
Bluetooth	V4.2 – Supports BLE and Classic Bluetooth

4.2 ESP32-CAM Camera Module

The ESP32-CAM is a feature-rich camera module based on the ESP32 chip. It is compact in size and consumes low power, making it suitable for various applications. The module is equipped withan OV2640 camera sensor and includes a TF card slot for convenient storage of captured images orvideos. One notable feature of the ESP32-CAM is its 4MB PSRAM, which enables efficient buffering of images for tasks such as video streaming, preventing crashes or memory limitations.

With this module, you can capture higher-quality pictures without compromising the performanceof the ESP32. Additionally, the ESP32-CAM offers an onboard LED for flash functionality and

multiple GPIO pins for easy connection to peripherals, expanding its versatility and potential.

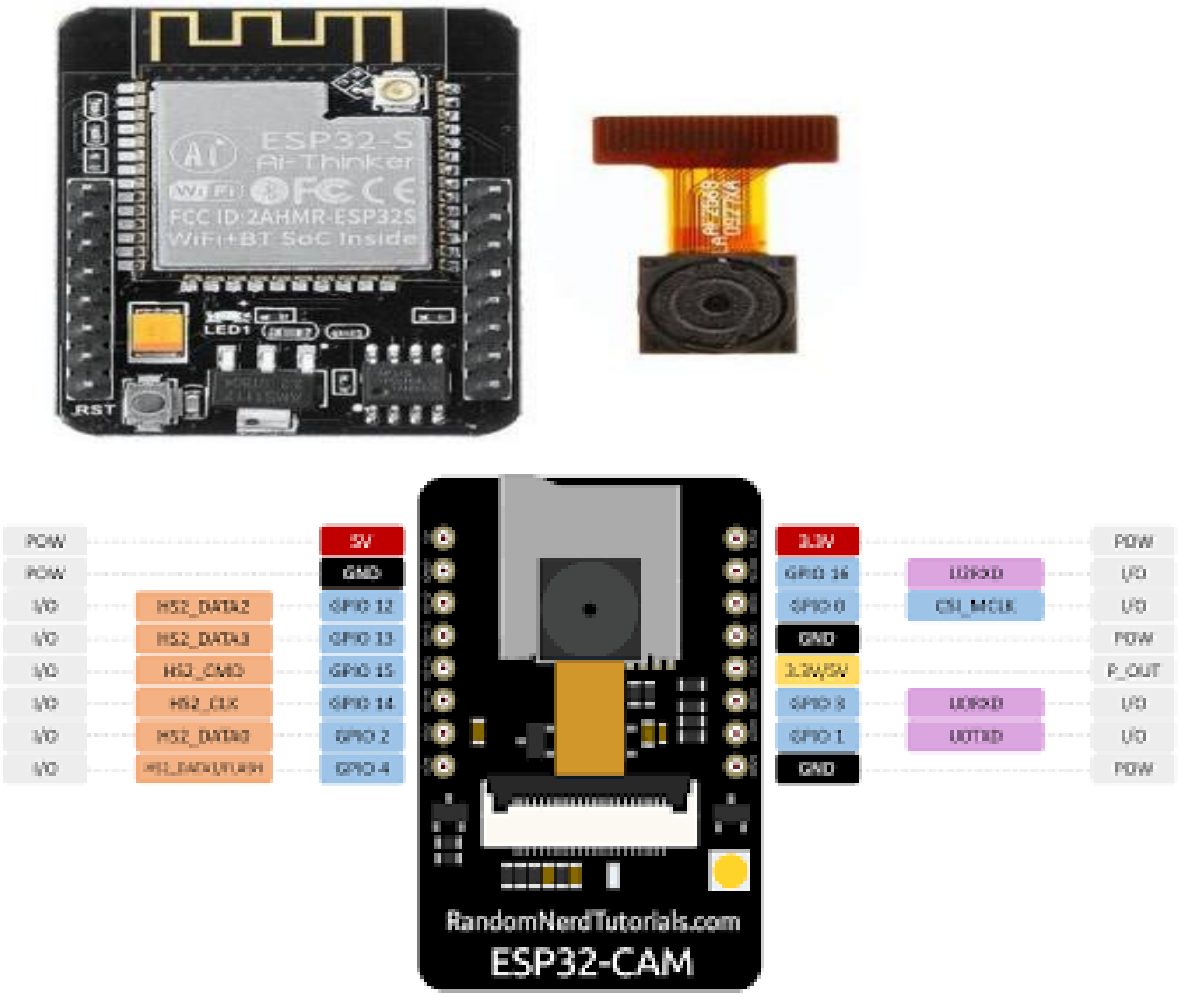


Fig 4.2: ESP-32 CAM

4.2.1 Features

• The ESP32-S module, which is integrated on the board, provides support for both Wi-Fi and Bluetooth connectivity.

• The board features an OV2640 camera with a flash, enabling image and video capture.

• An onboard TF card slot is available, allowing for data storage of up to 4GB on a TF card.

• The module supports Wi-Fi video monitoring and Wi-Fi image upload, enabling remote access andsharing of captured media.

• Multiple sleep modes are supported, with the deep sleep mode consuming as low as 6mA of current.

• The control interface of the board is easily accessible via a pin header, facilitating integration andembedding into various user products.

CHAPTER - 5

DESIGN AND IMPLEMENTATION

The smart door lock system is designed to enhance the security of homes and offices by using moderntechnology. Traditional locks are often prone to vulnerabilities like lock picking, key duplication,and unauthorized access. The smart door lock system overcomes these limitations by providing a more secure and convenient way of locking and unlocking doors. The system is integrated with AWS services like AWS IoT Core, AWS Lambda, S3 bucket, and AWS Rekognition to provide a reliableand efficient way of detecting and recognizing authorized users. The system also provides real-time notifications to the authorized user's smartphone through the Telegram app, enabling them to monitorand control the lock status from anywhere. The smart door lock system is a reliable and efficient solution to enhance the security of homes and offices.

In this report two projects are covered:

1. Smart door lock system using ESP32-CAM and Telegram.
2. Smart door lock system using ESP32-CAM and AWS.

5.1 Smart door lock system using ESP32-CAM and Telegram.

The smart home system using ESP32-CAM and Telegram is a project aimed at revolutionizing homeautomation and control. With the increasing demand for convenience and efficiency in our daily lives, this system offers a seamless solution to manage home devices remotely. By leveraging the power of the ESP32-CAM module and the popular messaging platform Telegram, users can effortlessly control and monitor their appliances from anywhere, at any time. The system incorporatesfeatures such as real-time notifications, energy optimization, and secure cloud integration, ensuring a user-friendly and intelligent home management experience. With this innovative solution, users can enhance their living environment, improve energy efficiency, and enjoy the convenience of a smart home ecosystem.

5.1.1 System Architecture:

5.1.1.1 User Access:

- User can access the smart door lock system through smartphone with internet.
- User interface for our system is the Telegram messaging app.

- By using Telegram, users can easily interact with the smart door lock system from anywhere.
- Established communication between Telegram and AWS.
- Send commands and receive real-time updates the smart door lock.

5.1.1.2 Telegram-bot:

- Telegram bot allowing data to be transferred and commands to be executed seamlessly between ESP32-CAM and user.
- Create a Telegram bot using the BotFather.
- Generate the API token using BotFather.
- This API token serves as an authentication mechanism for bot to access the Telegram API.
- you would typically use the Telegram Bot API's “sendMessage” method to transfer data

5.1.1.3 Telegram Bot Interaction:

- The telegram bot process user commands and coordinates the actions of the smart door lock.
- User sends a command to bot, the bot analyzes its content.
- The bot uses predefined commands and keywords to determine the action.
- The bot sends the instructions and data to the ESP32-CAM.
- The ESP32-CAM module receives the instructions and performs the actions .
- Two-way communication.

5.1.1.4 Feedback to User:

- Feedback to the user is a critical aspect of our smart door lock system.
- After receiving and processing commands, the Telegram bot sends a response message to the user.
- The response message serves various purposes, depending on the executed command.
- For instance, if the user requested to unlock the door, the bot confirms the successful unlocking and notifies the user.
- Similarly, if the user requested to check the door status, the bot provides real-time information on whether the door is locked or unlocked.
- In addition to textual feedback, the system can also send images captured by the ESP32-CAM module to the user.

- This feature allows users to visually monitor their door or capture evidence in case of any security concerns.
- The feedback provided by the system ensures that users stay informed about the actions performed and the current state of the door lock system.

5.1.1.5 End-to-End Security:

- Ensuring the security of our smart door lock system is of utmost importance.
- To achieve end-to-end security, we have implemented several measures.
- Firstly, the communication between the Telegram bot and the ESP32-CAM module is encrypted using secure protocols.
- This prevents unauthorized access or tampering with the command data during transmission.
- Additionally, user authentication and authorization mechanisms are in place.
- Users need to authenticate themselves before gaining access to the system, adding an extra layer of security.
- The system verifies user credentials or authentication tokens to ensure authorized access.
- Furthermore, the ESP32-CAM module itself has built-in security features, such as access control and device-level encryption.
- These measures collectively safeguard our smart door lock system against potential security vulnerabilities and unauthorized access attempts.
- By prioritizing end-to-end security, we ensure that users can trust the reliability and integrity of our smart door lock system.

5.2 Smart door lock system using ESP32-CAM and AWS.

The process taking place during the project is ESp-32 cam will click a photo of a person at the door andalso make a video clip of a few seconds that photo and clip will be transferred to the ESp-32 Wi-Fi module, it will send that photo to AWS using AWS Lambda and 4. AWS Rekognition will analyze the photo i.e., match the face with already stored faces in the AWS S3 bucket by the user.

If the face matches, then it will send an SOS with the information of the person at the door to the user and ask permission to unlock the lock of the door. Now if the face does not match then it will send an alert to the user with the recent video clip to the user and also ask permission to unlock the door or not.

5.2.1 System Architecture

The system architecture for the smart door lock system using AWS is designed to provide secure and efficient access control with the integration of various components. The architecture encompasses the ESP32-CAM module, AWS services, and the Telegram app, enabling seamless communication and control. At the core of the system is the ESP32-CAM module, which serves as the primary device responsible for capturing images, processing data, and interacting with the AWS cloud. The ESP32-CAM module is equipped with a camera module, microcontroller, and Wi-Fi connectivity, allowing it to capture real-time images and communicate with the AWS cloud services. The AWS services utilized in the architecture include AWS IoT Core, AWS Lambda, and AWS S3. AWS IoT Core provides a secure and scalable infrastructure for device communication and management. The ESP32-CAM module connects to AWS IoT Core, enabling bidirectional communication between thedevice and the cloud.

AWS Lambda, a serverless computing service, is used for processing and analyzing the image data received from the ESP32-CAM module. It hosts face detection and recognition algorithms, leveraging the power of AWS Rekognition for accurate and efficient identification of individuals. Additionally, AWS S3 (Simple Storage Service) is utilized for storing and retrieving images and other data related to the smart door lock system. It provides a reliable and scalable storagesolution for the captured images and other relevant information. The Telegram app serves as the userinterface and control mechanism for the smart door lock system. It allows users to interact with the system, receive notifications, and control access to the door. Through the Telegram app, users can send commands, receive status updates, and even view captured images or recognized faces. The system architecture ensures seamless integration and communication between the ESP32-CAM module, AWS services, and the Telegram app. It provides a robust and secure platform for efficientaccess control and user interaction.

By implementing this system architecture, the smart door lock system using AWS achieves enhanced security, real-time face detection and recognition, and convenient user control, making it a reliable and efficient solution for access control in various applications.

5.2.2 Hardware Setup

The hardware setup for the smart door lock system using AWS involves the integration of various components to enable seamless communication and operation. The following hardware componentswere utilized in the implementation:

- **ESP32-CAM Module:** The ESP32-CAM module serves as the central device for capturing images, processing data, and communicating with AWS services. It incorporates a camera module, a powerful microcontroller, and Wi-Fi connectivity, making it an ideal choice for this project.
- **Electric Door Lock:** An electric door lock mechanism is used to control the physical accessto the door. The electric lock is connected to the ESP32-CAM module, allowing it to lock and unlock the door based on user commands.
- **Power Supply:** A stable and reliable power supply is essential for the proper functioning ofthe system. The ESP32-CAM module and electric door lock are powered by an appropriate power source, ensuring continuous operation.
- **Internet Connectivity:** The hardware setup requires a stable internet connection to establishcommunication with AWS services. This can be achieved through a Wi-Fi network or other reliable internet connectivity options.
- **Optional Components**: Depending on specific requirements, additional components such asLEDs, buttons, and sensors can be incorporated into the hardware setup to enhance functionality and user experience. These components can be used for status indication, user interaction, or environmental sensing if needed.

The hardware setup should be carefully executed, ensuring proper connections, power supply, and compatibility between the components. It is essential to follow the provided documentation and guidelines for each component to ensure a successful implementation.

5.2.3 ESP32 – Wi-Fi Module: The ESP32 is a very versatile System On a Chip (SoC) that can be used as a general-purpose microcontroller with quite an extensive set of peripherals including Wi-Fi and Bluetooth wireless capabilities. ESP32 is a single-chip 2.4 GHz Wi-Fi and Bluetooth combo chip designed with TSMC ultra-low power 40 nm technology. It is designed and optimized for the best power performance, RF performance, robustness, versatility, features and reliability, for a wide variety of applications, and different power profiles.

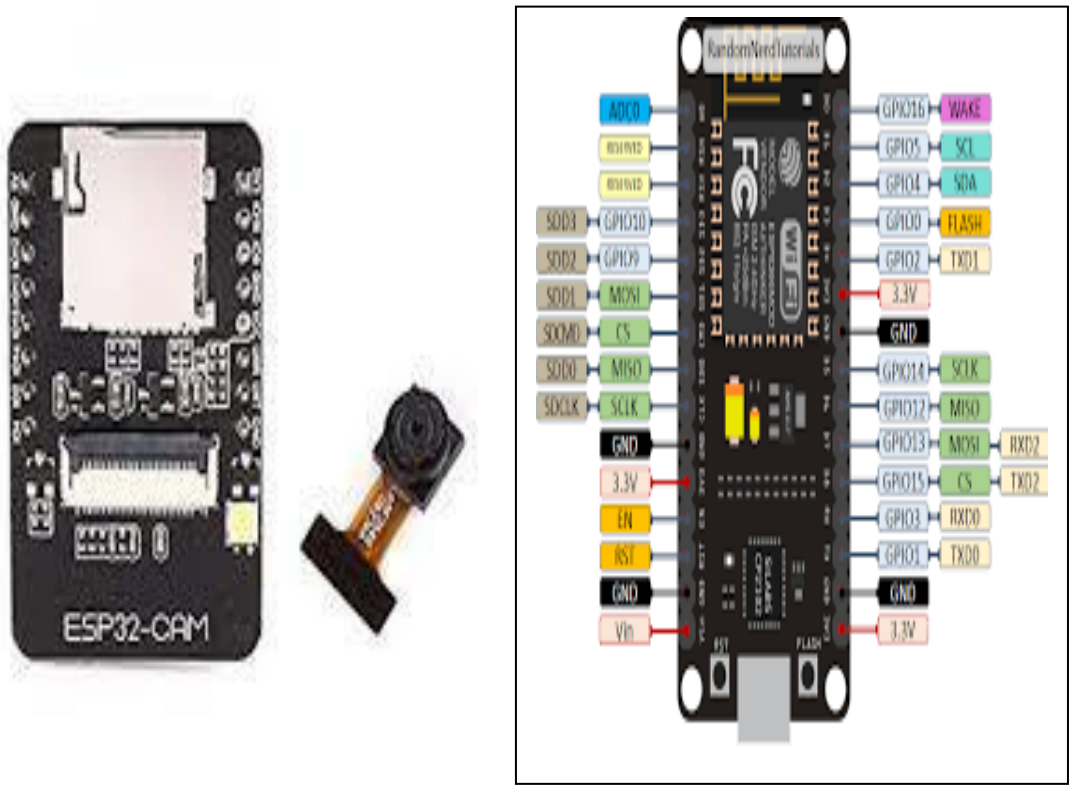


Fig 5.1 ESP-32 Wi-Fi Module

5.2.4 ESP32-CAM Camera Module

The ESP32-CAM is a small-size, low-power camera module based on ESP32. It comes with an OV2640 camera and provides an onboard TF card slot. This board has 4MB PSRAM which is usedfor buffering images from the camera into video streaming or other tasks and allows you to use higherquality in your pictures without crashing the ESP32.

5.2.4.1 Features

• The ESP32-S module, which is integrated into the board, provides support for both Wi-Fi and Bluetooth connectivity.

• The board features an OV2640 camera with a flash, enabling image and video capture.

• An onboard TF card slot is available, allowing for data storage of up to 4GB on a TF card.

• The module supports Wi-Fi video monitoring and Wi-Fi image upload, enabling remote access andsharing of captured media.
• Multiple sleep modes are supported, with the deep sleep mode consuming as low as 6mA of current.
• The control interface of the board is easily accessible via a pin header, facilitating integration andembedding into various user products.

5.2.5 Electric Door Lock

The electric door lock is a crucial component of the smart door lock system, responsible for controlling the physical access to the door. It operates based on commands received from the ESP32-CAM module, allowing users to lock and unlock the door remotely using the Telegram app. The electric door lock used in this project is a secure and reliable locking mechanism designed for smart home applications. It offers the following features:

Robust Construction: The electric door lock is constructed using durable materials to ensure longevity and withstand regular use. It is designed to provide reliable locking and unlocking operations.
Secure Locking Mechanism: The electric door lock employs a secure locking mechanism, offering enhanced protection against unauthorized access. It utilizes advanced locking mechanisms such as electromagnetic or motor-driven locks, ensuring the door remains securely locked when required.
Remote Control Capability: The electric door lock is integrated with the ESP32-CAM module, enabling remote control via the AWS cloud and Telegram app. Users can conveniently lock or unlock the door from anywhere, providing flexibility and convenience.
Power Efficiency: The electric door lock is designed to be power-efficient, consuming minimal power during operation. It ensures optimal energy consumption and contributes to the overall efficiency of the smart door lock system.
Compatibility: The electric door lock is compatible with the ESP32-CAM module and can be easily integrated into the hardware setup. It is designed to work seamlessly with the system components, ensuring smooth operation and reliable functionality.

When implementing the electric door lock, it is essential to follow the manufacturer's guidelines and wiring instructions. Proper installation and secure mounting are necessary to ensure the reliable operation and security of the smart door lock system.

The integration of the electric door lock into the system enables secure access control, allowing users to conveniently lock and unlock their doors remotely. With its robust construction, remote control capabilities, and power efficiency, the electric door lock enhances the overall functionality and security of the smart door lock system.

5.2.6 Software Implementation

The software implementation of the smart door lock system plays a vital role in facilitating the communication between the hardware components, data processing, and user interaction. It involvesthe development and integration of various software components to ensure the smooth operation andfunctionality of the system.

In this project, the following software components were implemented:

Firmware Development: The firmware for the ESP32-CAM module was developed to handle the communication with AWS and perform essential tasks such as capturing images, processing data, and controlling the electric door lock. The firmware was written in C++ programming language, utilizing libraries specific to the ESP32 platform.

AWS Integration: The smart door lock system utilizes AWS services for data transfer, storage, and processing. AWS IoT Core was used to establish the MQTT communication between the ESP32-CAM module and AWS cloud. AWS Lambda functions were developedto process the incoming data, perform face detection and recognition, and trigger appropriate actions.

Face Detection and Recognition: OpenCV, an open-source computer vision library, was used for implementing the face detection and recognition functionality. The captured imagesfrom the ESP32-CAM module were processed using OpenCV algorithms to detect and recognize faces. AWS Rekognition, a powerful machine learning service, was employed to further enhance the face recognition capabilities.

Telegram Integration: The Telegram API was utilized for integrating the smart door lock system with the Telegram app. A Telegram bot was created and configured to receive commands and notifications from users. The bot was programmed to interpret the user commands, communicate with AWS services, and control the electric door lock accordingly.

User Interface: The user interface was designed and implemented using a combination of Telegram app and custom commands. Users can interact with the system by sending commands to the Telegram bot, such as "Lock" or "Unlock." The system responds to these commands and provides real-time status updates and notifications to the users.

The software implementation involved extensive coding, testing, and integration to ensure the smooth and reliable operation of the smart door lock system. Various software development tools, programming languages, and libraries were utilized to achieve the desired functionality and seamlessintegration with the hardware components.

By implementing the software components described above, the smart door lock system enables secure and convenient access control. The integration with AWS, face detection and recognition algorithms, and Telegram app provides a user-friendly interface and advanced security features.

5.2.7 Telegram Application

The integration of the Telegram application plays a crucial role in providing a user-friendly interfacefor controlling and monitoring the smart door lock system. Telegram, a popular instant messaging platform, offers a robust API that enables seamless communication between the user and the system.

In this project, the Telegram application was used to enable remote control and real-time notificationsfor the smart door lock system. The following features were implemented:

Telegram Bot: A Telegram bot was created to act as an intermediary between the user and the smart door lock system. The bot handles incoming commands and messages from users, processes them, and triggers the corresponding actions.

User Commands: Users interact with the system by sending commands to the Telegram bot. Common commands include "Lock" to lock the door, "Unlock" to unlock the door, and "Status" to check the current status of the door. These commands are interpreted by the bot and relayed to the system for execution.

Notifications: The Telegram bot sends real-time notifications to the users regarding various events and activities related to the smart door lock system. For example, when someone successfully unlocks the door or when an unrecognized face is detected, a notification is sentto the authorized users through the Telegram app.

Secure Communication: The communication between the Telegram app and the smart door lock system is secured using encryption and authentication mechanisms provided by the Telegram API. This ensures the privacy and integrity of the data transmitted between the user and the system.

By integrating the Telegram application, users can conveniently control and monitor the smart

door lock system from anywhere using their smartphones. The Telegram bot acts as a virtual assistant, providing a simple and intuitive interface for managing access to the door and receiving real-time notifications.

The Telegram application, with its extensive features and user-friendly interface, enhances the accessibility and usability of the smart door lock system. It allows users to have remote control over the door lock and stay informed about the system's status and activities in real- time.

5.2.8 AWS Integration

The integration of Amazon Web Services (AWS) in the smart door lock system adds scalability, reliability, and advanced functionalities to the overall system architecture. AWS offers a comprehensive suite of cloud services that enable secure storage, data processing, and real-time communication.

In this project, AWS was utilized for various purposes, including:

I. AWS IoT Core

AWS IoT Core is a managed cloud service that enables secure and scalable communication between connected devices (such as the ESP32-CAM) and the cloud. It provides a reliable and secure infrastructure for managing IoT devices, message routing, and device authentication.

In the smart door lock system, AWS IoT Core played a crucial role in establishing a secure and bi-directional communication channel between the ESP32-CAM module and the cloud. Here's how AWS IoT Core was utilized:

Device Connectivity: The ESP32-CAM module was configured as an IoT device and registered with AWS IoT Core. It connected to AWS IoT Core using the MQTT (Message Queuing Telemetry Transport) protocol, ensuring lightweight and efficient communication.

Device Authentication and Security: AWS IoT Core provides device authentication and security mechanisms to ensure that only authorized devices can connect and communicatewith the cloud. The ESP32-CAM was assigned a unique device certificate, allowing it tosecurely authenticate and establish a trusted connection with AWS IoT Core.

Message Broker: AWS IoT Core acts as a message broker, facilitating the bi-directional communication between the ESP32-CAM and other cloud services. It receives messages published by the ESP32-CAM and forwards them to the appropriate destinations, such asAWS Lambda or AWS S3.

Device Shadow: AWS IoT Core offers a device shadow feature, which provides a virtual

representation of the device's state and allows for synchronization and control of the device's properties. The device shadow was used to store and retrieve the current status of the smart door lock system, enabling remote monitoring and control.

Rule Engine: AWS IoT Core provides a rule engine that allows for processing and routingof incoming messages based on predefined rules. The rule engine was utilized to trigger specific actions based on the received messages from the ESP32-CAM, such as invokingAWS Lambda functions for further data processing.

By leveraging AWS IoT Core, the smart door lock system benefits from a secure, scalable,and managed infrastructure for IoT device connectivity. The integration with AWS IoT Core ensures reliable and efficient communication between the ESP32-CAM and the cloud, enabling seamless data transfer, device management, and remote control capabilities.AWS **Lambda:** AWS Lambda, a serverless compute service, was used to implement serverless functions that process the data received from the smart door lock system. For example, the Lambda function can analyze the incoming data for face recognition or perform additional logic based on predefined rules.

II. **AWS S3:**(Simple Storage Service)

AWS S3 is a scalable and highly available object storageservice offered by Amazon Web Services. It provides a secure and durable platform for storing and retrieving data, making it an ideal choice for our smart door lock system. Here's how AWS S3 was utilized:

Data Storage: AWS S3 was used to store various types of data generated by the smartdoor lock system, including images captured by the ESP32-CAM module, logs, and **configuration files**. The S3 bucket acted as a centralized storage repository for thesefiles, ensuring durability and easy access.

Image Storage: Captured images from the ESP32-CAM module were securely uploaded to the designated S3 bucket. This allowed for convenient storage, retrieval,and management of the images, enabling future analysis, auditing, and access control.

Data Backup and Redundancy: AWS S3 offers built-in redundancy and datadurability features. The data stored in the S3 bucket is automatically replicated acrossmultiple AWS Availability Zones, ensuring high availability and data resiliency. Thisprovided an additional layer of data protection for the smart door lock system.

Access Control: AWS S3 allows fine-grained access control to the stored data. Access permissions and policies were defined to ensure that only authorized users orservices could interact with the S3 bucket. This helped to protect the privacy and integrity of the data stored within the bucket.

Integration with Other AWS Services: The S3 bucket seamlessly integrated with other AWS services. For example, AWS Lambda functions were triggered when newimages were uploaded to the S3 bucket, allowing for real-time processing and analysisof the captured images. Additionally, the S3 bucket could be used as a source for generating notifications or triggering other AWS services.

By utilizing AWS S3, the smart door lock system benefited from a highly scalable, secure, and reliable storage solution. The integration with AWS S3 allowed for efficient and seamlessstorage of data generated by the system, providing a solid foundation for data analysis, backup, and integration with other AWS services.**AWS Rekognition:** AWS Rekognition, a powerful image analysis service, was employed for face detection and recognition capabilities. The captured images from the ESP32-CAM were sent to AWS Rekognition for processing and identification of authorized users, allowing for seamless access control.

III. AWS Lambda:

It is a serverless compute service provided by Amazon Web Services. It allows the executionof code in response to events without the need for managing servers or infrastructure. In oursmart door lock system, AWS Lambda played a crucial role. Here's how AWS Lambda was utilized:

Event-Driven Execution: AWS Lambda was used to execute code functions in response to events or triggers. For example, when new images were uploaded to the S3 bucket, a Lambda function was triggered to perform real-time image processing and facial recognition.

Image Processing and Facial Recognition: The Lambda function implemented image processing algorithms and utilized AWS Rekognition, a powerful imageanalysis service, to perform facial recognition on the uploaded images. This allowed for the identification of authorized individuals and helped enhance the security of thesmart door lock system.

Serverless Architecture: AWS Lambda's serverless architecture eliminated the needfor managing and scaling servers. It automatically scaled the execution environment based on the incoming workload, ensuring optimal performance and cost-efficiency for the smart door lock system.

Integration with Other AWS Services: AWS Lambda seamlessly integrated with other AWS services, enabling a robust ecosystem for the smart door lock system. Forinstance, the Lambda function could interact with AWS S3 for image retrieval, AWSIoT Core for device control, and AWS DynamoDB for storing and retrieving access control policies.

Customizable and Configurable: AWS Lambda provided the flexibility to write custom code functions in various programming languages, including Python and Node.js. This allowed for the implementation of specific logic and business rules tailored to the smart door lock system requirements.

By using AWS Lambda, our smart door lock system achieved a scalable and event- driven architecture. The serverless nature of Lambda simplified deployment and management, while its seamless integration with other AWS services enhanced the system's functionality and flexibility. The utilization of AWS Lambda enabled real-time image processing, facial recognition, and seamless interaction with other AWS components, makingit a critical component of our system's implementation.The integration of AWS into the smartdoor lock system adds a layer of intelligence and scalability. The cloud-based architecture allows for seamless integration of additional features and supports future expansion of the system. Leveraging AWS services, the smart door lock system benefits from high availability,security, and the ability to handle large-scale deployments.

IV. AWS Rekognition:

It is a powerful image and video analysis service provided by Amazon Web Services. It offersa wide range of computer vision capabilities, including facial analysis, object detection, and scene understanding. In our smart door lock system, AWS Rekognition played a significant role. Here's how AWS Rekognition was utilized:

Facial Recognition: AWS Rekognition's facial recognition feature was employed toidentify and authenticate individuals accessing the smart door lock system. By comparing the captured facial images with a pre-defined database of authorized users,Rekognition enabled accurate and reliable identification.

Facial Analysis: Rekognition provided advanced facial analysis capabilities, including emotion detection, age estimation, and gender identification. This functionality allowed for a deeper understanding of the captured images, facilitating enhanced security and user experience within the smart door lock system.

Real-time Image Processing: AWS Rekognition's real-time image processing capabilities were leveraged to perform instantaneous analysis of the images capturedby the ESP32-CAM. The captured images were sent to Rekognition for facial recognition and analysis, providing rapid and accurate results for access control.

Customizable Training: Rekognition offered the ability to train custom machine learning

models for specific use cases. In our project, we utilized this feature to traina custom facial recognition model using a dataset of authorized individuals. This allowed for improved accuracy and tailored identification within the smart door locksystem.

Integration with AWS Services: AWS Rekognition seamlessly integrated with otherAWS services, enabling a comprehensive ecosystem for our smart door lock system.For instance, the Rekognition results seamlessly integrated with AWS Lambda, allowing for real-time decision-making and control based on the facial recognition outcomes.

By incorporating AWS Rekognition into our smart door lock system, we achieved accurate and efficient facial recognition, enhancing the overall security and user experience. The advanced facial analysis capabilities provided valuable insights, while the real-time processing ensured prompt access control. The integration with other AWS services expandedthe functionality of our system and facilitated seamless interactions between different components. AWS Rekognition played a pivotal role in enabling robust facial recognition and analysis within our smart door lock system.

V. AWS CloudWatch:

It was utilized for monitoring and logging the system's performance and activities. It providesreal-time insights into system metrics, such as device connectivity, data transfer, and Lambdafunction execution, enabling efficient monitoring and troubleshooting.

By using the power of AWS, the smart door lock system achieves a robust and efficient architecture that ensures reliable connectivity, advanced data processing, and secure storage. The integration with AWS enhances the overall performance and functionality of the system, providinga seamless and secure user experience.

VI. Face Detection and Recognition

Face detection and recognition played a crucial role in our smart door lock system. By incorporatingadvanced computer vision techniques and leveraging AWS Rekognition, we were able to enhance the security and convenience of our system. Here's an overview of the face detection and recognitionprocess:

Face Detection: We utilized the ESP32-CAM module's capabilities along with OpenCV (Open-Source Computer Vision Library) to perform real-time face detection. The ESP32-CAM captured images using its built-in camera, and OpenCV algorithms were applied to detect faces within the images. This initial step allowed us to identify the presence of a personin front of the door.

Image Preprocessing: Before sending the captured images to AWS Rekognition, we performed the necessary preprocessing steps. This involved resizing, cropping, and optimizing the images to ensure optimal processing and accuracy during face recognition.

AWS Rekognition: We leveraged AWS Rekognition, a powerful image and video analysis service, for accurate face recognition. The preprocessed images were sent to Rekognition's APIs for facial feature extraction and comparison. Rekognition utilized sophisticated deep learning algorithms to identify facial landmarks, extract facial features, and create unique face signatures.

Face Comparison: In our system, we had a database of authorized users' face signatures stored in AWS. Rekognition compared the extracted face signature from the captured imagewith the stored face signatures. By matching key facial features and using similarity metrics, Rekognition determined the identity of the individual in the image.

Access Control and User Notification: Based on the results of the face comparison, our system made access control decisions. If the detected face matched an authorized user's face signature, the smart door lock system granted access. In case of a mismatch or unauthorized face, access was denied. Additionally, user notifications were sent via the Telegram app to provide real-time updates on access attempts and outcomes.

The integration of face detection and recognition capabilities added an extra layer of security and personalized access control to our smart door lock system. By leveraging the powerful algorithms and machine learning models of AWS Rekognition, we achieved accurate and reliable identification of individuals, ensuring that only authorized users could access the secured area.The combination of ESP32-CAM's real-time face detection, image preprocessing, and AWS Rekognition's advanced face recognition capabilities resulted in a robust and efficient face detectionand recognition system for our smart door lock project.

VII. User Interface and Control

The user interface and control aspect of our smart door lock system played a vital role in providing a seamless and convenient user experience. By integrating the Telegram app with our system and leveraging its extensive features, we were able to offer remote control and real-time notifications to the users. Here's an overview of the user interface and control components:

Telegram Integration: We chose the Telegram app as our user interface platform due to its wide availability, cross-platform compatibility, and rich set of features. Telegram provided a

secure and reliable communication channel between the users and the smart door lock system.

User Registration: Users were required to register with the Telegram bot associated with our system. During the registration process, users provided their essential details and authorized their Telegram accounts to interact with the smart door lock system.

Command-based Control: Through the Telegram app, users could send specific commandsto the smart door lock system. For instance, users could issue commands to lock or unlock the door, check the status of the door, or request access logs. These commands were processedby the system, and appropriate actions were taken accordingly.

Real-time Notifications: Our system sent real-time notifications to the users via the Telegram app. Users received notifications when someone attempted to access the door, whenthe door was successfully locked or unlocked, and in case of any system events or errors. Thisensured that users remained informed about the status and activities of the smart door lock system, even when they were away.

Access Logs and History: The Telegram app allowed users to access logs and history relatedto the smart door lock system. Users could review access logs to track who accessed the doorand when. This feature provided an added layer of security and accountability.

By integrating the Telegram app as the user interface and control platform, we offered users a convenient and intuitive way to interact with the smart door lock system remotely. The seamless integration allowed users to control the door, receive real-time notifications, and access system logsand history, all from the convenience of their smartphones.

The Telegram integration provided a user-friendly and accessible interface, enhancing the overall functionality and usability of our smart door lock system. Users could securely control and monitor their doors, ensuring the security and convenience of their premises.

User Registration: Users were required to register with the Telegram bot associated with our system. During the registration process, users provided their essential details and authorized their Telegram accounts to interact with the smart door lock system.

Command-based Control: Through the Telegram app, users could send specific commandsto the smart door lock system. For instance, users could issue commands to lock or unlock the door, check the status of the door, or request access logs. These commands were processedby the system, and appropriate actions were taken accordingly.

5.3 SYSTEM WORKFLOW:

I have divided my project into 4 phases -:

5.3.1 Establishing Connection (ESP32-CAM and AWS)

5.3.2 Data Transfer (ESP32-CAM and AWS)

5.3.3 Face Detection and Recognition (AWS)

5.3.4 User Notification and Control (Telegram Integration)

5.3.1 Establishing Connection (ESP32-CAM and AWS) :

Step 1. Create a thing in AWS.

Step 2. Generating a certificate.

Step 3. Attaching a policy to it.

Step 4. Use generated endpoint for communication.

Step 5. Subscribe and Publish MQTT topic.

Step 6. – Installing ESP8266 sketch data upload tool in Arduino IDE.

Step 7. – Modifications in the Arduino sketch according to the thing.

Step 8. – Uploading AWS certificates & code to the NodeMCU.

Step 9. – Testing/Subscription of things on Amazon Web Services.

Step 10. Use generated thing name, certificate, endpoint, subscribe and publish topic in code andburn it in ESP32- Cam module.

Step 11. – Live demo of Data Logging.

Screenshots:

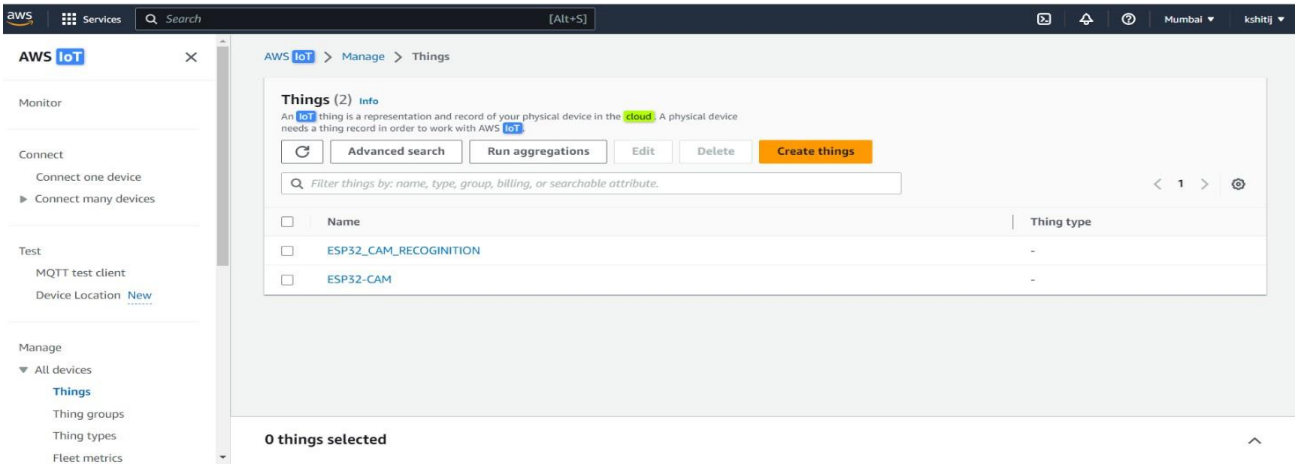


Fig 5.2 Create Thing at AWS

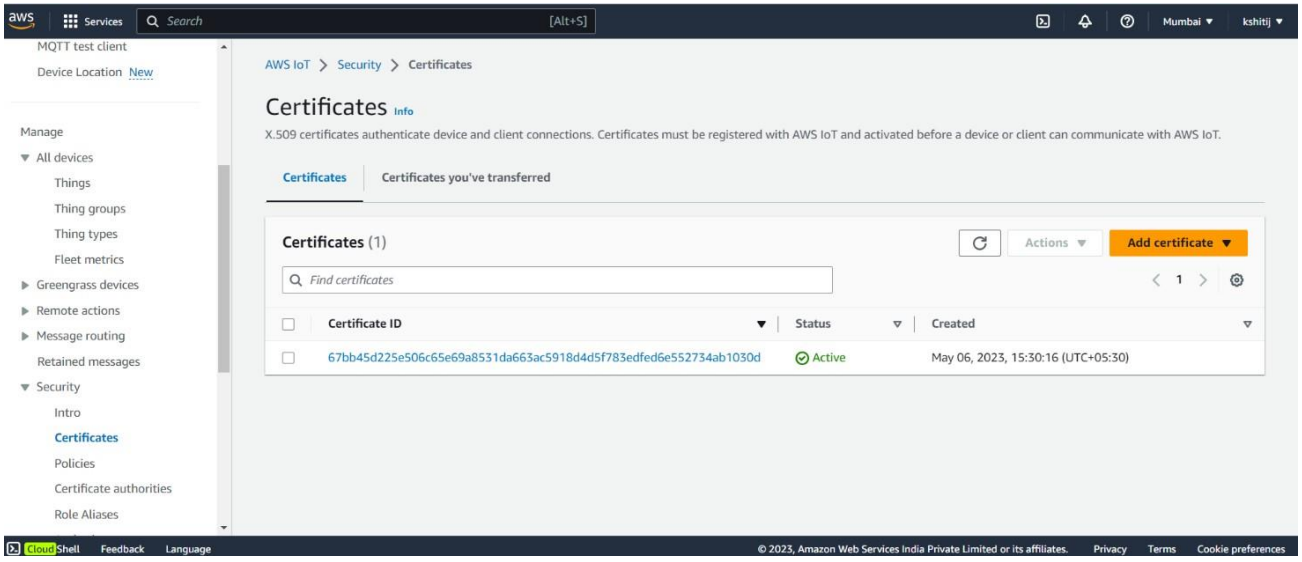


Fig. 5.3 Create certificates

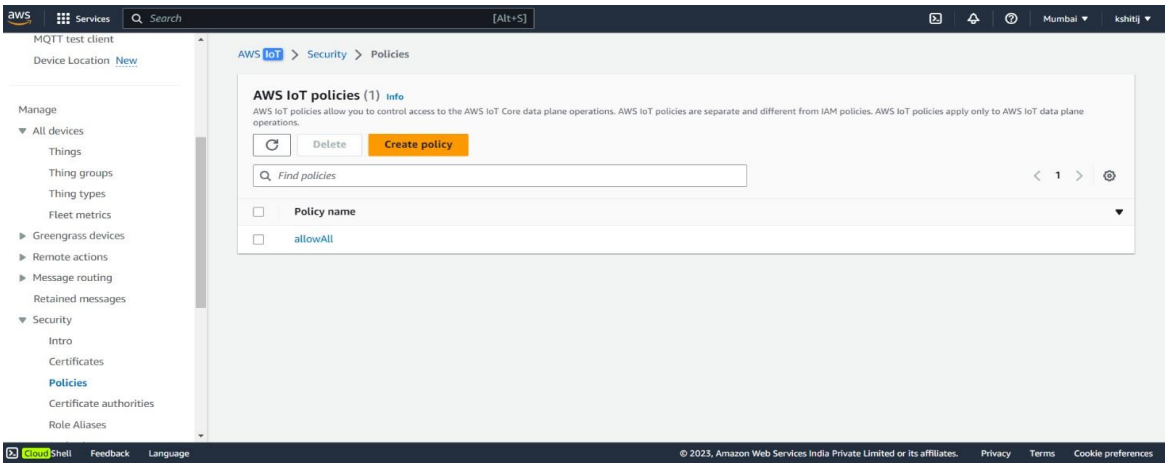


Fig. 5.4 Create policy

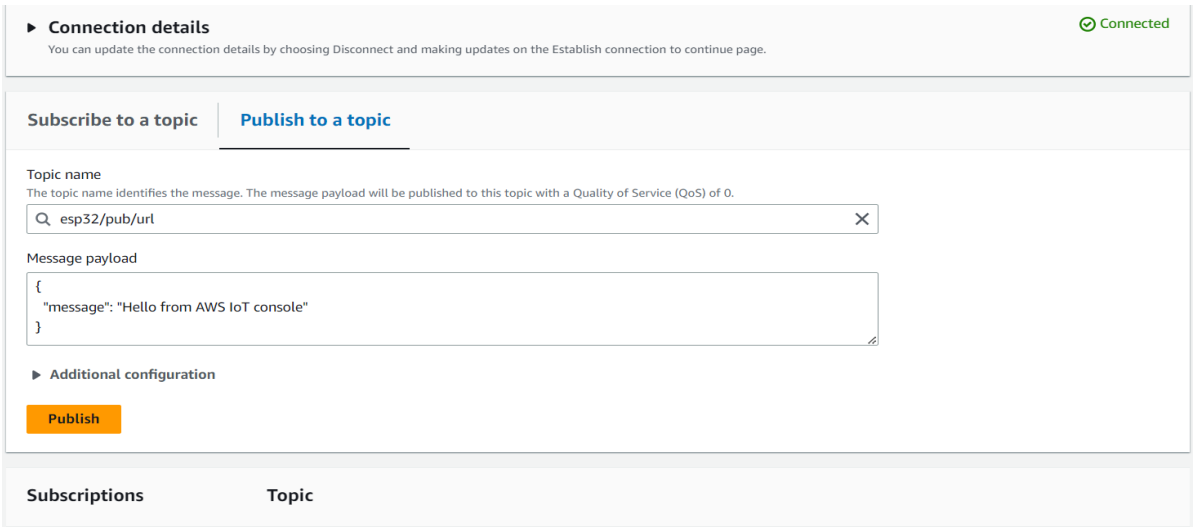


Fig. 5.5 Publish Topic

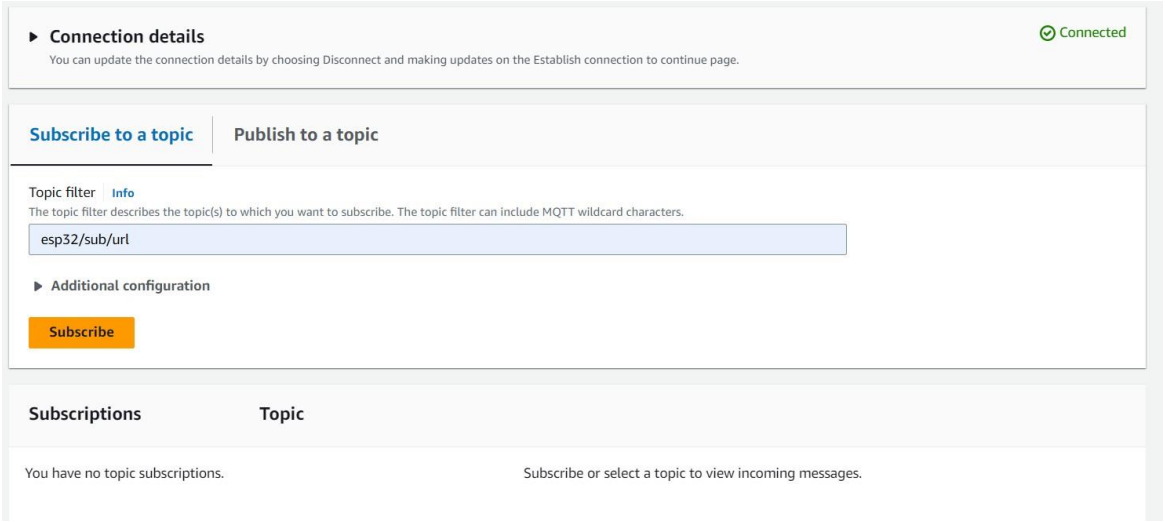


Fig. 5.6 Subscribe Topic

5.3.2 Data Transfer (ESP32-CAM and AWS):

Step 1. Create an S3 bucket.

Step 2. Create a Lambda function, that generates a signed URL to upload images over the S3 bucket.

Step 3. To transfer data, generate message rule.

Step 4. Whenever something publishes over the topic, the lambda function trigger.

Step 5. Then signed URL will be generated.

Step 6. Image transfer through that URL over the S3 bucket.

5.3.2 Face Detection and Recognition Workflow

Face Detection Workflow:

Step 1. ESP32-CAM captures an image of the person at the door.

Step 2. The image is sent to AWS S3.

Step 3. Rekognition analyzes the image and identifies any faces present.

Step 4. The attributes are extracted for further processing.

Face Recognition Workflow:

Step 1. The detected faces are compared against a face collection in AWS Rekognition.

Step 2. AWS Rekognition uses a face recognition algorithm.

Step 3. If a match is found, the system identifies the person and retrieves their stored information.

Step 4. The person's name and information are sent as a notification to the user's Telegram app.

Step 5. Providing “Token API” of telegram bot to lambda function where data to be send by AWS.

5.3.4 User Notification and Control:

Step 1. Create a Telegram bot using the BotFather

Step 2. Generate the API token using BotFather

Step 3. Set up an AWS API Gateway.

Step 4. Create a Lambda function in AWS.

Step 5. Telegram Bot API within the Lambda function to send data back to the user.

CHAPTER – 6
CONCLUSION

In conclusion, the development of the smart door lock system using ESP32-CAM and Telegram hasproven to be a successful venture. The project aimed to enhance home security and provide remote control access to the door lock through a user-friendly interface. Through the implementation of theESP32-CAM module, image processing capabilities were leveraged for face detection and recognition. The integration with Telegram allowed users to conveniently control the door lock system from their mobile devices. Overall, the system demonstrated efficient communication between the ESP32-CAM, the Telegram app, and the door lock mechanism. The project achieved itsobjectives of improving security and providing convenient access control. Further improvements could include refining the face recognition algorithm and enhancing the system's robustness againstpotential security threats.

The development of the smart door lock system using ESP32-CAM, AWS, and Telegram as the user interface has yielded significant advancements in home security and remote accessibility. By integrating the ESP32-CAM module with AWS services such as IoT Core, Lambda functions, and S3 buckets, a robust and scalable infrastructure was established. The system successfully leveraged AWS Rekognition for efficient face detection and recognition, enabling authorized individuals to securely unlock the door using the Telegram app. The implementation showcased the potential of cloud-based solutions for IoT applications, providing seamless connectivity and reliable data transfer. However, further enhancements can be made to optimize thesystem's performance and improve its response time. Overall, the project has demonstrated the effective integration of ESP32-CAM, AWS, and Telegram to create an intelligent and secure door lock system with immense potential for broader IoT applications.

REFERENCES

[1] Sehoon Kim, Jin-Young Hong, Seil Kim, Sung-Hoon Kim, JunHyung Kim, Jake

Chun (2014). RESTful Design and Implementation of Smart Appliances for Smart Home, IEEE 11th Intl Conf onAutonomic and Trusted Computing, Vol. 3, Issue 4, April 2014, pp. 717 - 772

[2] Il-Kyu Hwang, Member, and Jin-Wook Baek (2007). Wireless Access Monitoring and Control System based on Digital Door Lock, IEEE Transactions on Consumer Electronics, Vol. 53, Issue. 4,
Nov 2007,pp. 1724 – 1730

[3] Subhankar Chattoraj, (2015). Home Automation Based on Different Sensors and Arduino as themaster controller, International Journal of Scientific and Research Publications, Vol. 5, Issue 10,
October 2015,pp. 1 – 4

[4] From sinarharian homepage, Title: Warga Emas Cedera Disamun Tiga Lelaki, URL
:http://www.sinarharian.com.my/wawancara/warga-emas-cederadisamun-tiga-lelaki-1.475210 [Data accessed: 8 March 2016]

[5] From sinarharian homepage, Title: Wanita warga emas cedera, rugiRM18,000 disamun, URL
:http://www.sinarharian.com.my/sukan/wanita-warga-emas-cederarugi-rm18-000-disamun- 1.253326 [Data accessed: 8 March 2016]

[6] Chunlai Zhou, Wenhui Huang, and Xiaoyun Zhao (2013), Study on architecture of smart home management system and key devices, Computer Science and Network Technology (ICCSNT), 20133rd International Conference, 12-13 October 2013, Dalian, pp. 1255 –1258

[7] Hsien-Tang Lin (2012), The Development of Control and Energy Usage Information Modules for Smart Homes, International Conference on Control, Automation and Information Sciences (ICCAIS), 26-29 November 2012, pp. 236 – 240

[8] Lin Liu, Yang Liu, Lizhe Wang, and Albert Zomaya (2015), Economical and Balanced Energy Usage in the Smart Home Infrastructure: A Tutorial and New Results, Iee Transactionson (**The Transactions on) Emerging Topics in Computing, Vol. 3,Issue 4, 4 December 2015, pp. 556 – 570

[9] YAN Wenbo, WANG Quanyu, and GAO Zhenwei (2015), Smart Home Implementation Based on Internet and WiFi Technology, Proceedings of the 34th Chinese Control Conference, 28 – 30 July2015, pp. 9072 – 9077

[10] R. Chutia, D. Sonowal and S. Sharma (2011), Remote Household Appliance Control System Using GSM, Proc. of the International Conference on Advanced Computing and Communication Technologies (ACCT 2011), pp. 540 – 542

[11] Z. Ghrairi, K.A. Hribernik, C. Hans, K.-D. Thoben (2012), Intelligent wireless communication devices for efficient data transfer and machine control, International Conference on Communications,
Computing and Control Applications (CCCA), 6 – 8 December 2012, pp. 1 – 6

[12] N.H. Ismail, Zarina Tukiran,N.N. Shamsuddin and E.I.S Saadon (2014), Android-based Home Door Locl(s Application via Bluetooth for Disabled People, International Conference on Control System, Computing and Engineering, 28 - 30 November 2014, pp.227 – 213

[13] Basma M. Mohammad El-Basioni, Sherine Mohamed Abd El-Kader, and Hussein S. Eissa (2014), Independent Living for Persons with Disabilities and Elderly People Using Smart Home
Technology International Journal of Application or Innovation in Engineering

Management(IJAIEM), Vol. 3, Issue 4, April 2014, pp. 11 – 28

[14] ZHENG Hong, Haiteng Zhang, PAN Li (2014). Modeling and Analysis of ZigBee Based Smart Home System, International Conference on Digital Home, 28 -30 November, pp. 242 – 245

[15] Dimitar H. Stefanov, Zeungnam Bien, and Won-Chul Bang (2004).The Smart House for OlderPersons and Persons with Physical Disabilities: Structure, Technology Arrangements, and Perspectives, IEEE Transactions On Neural Systems and Rehabilitation Engineering, Vol. 12, Issue.2, June 2004, pp. 228 – 250

[16] Raafat Aburukba, A. R. Al-Ali, Nourhan Kandil, and DialaAbuDamis (2016). Configurable ZigBee-based Control System for People with Multiple Disabilities in

Smart Homes, 2016 International Conference on Industrial Informatics and Computer Systems (CIICS), March 2016, pp.1 – 5

[17] Bessam Abdulrazak, Mounir Mokhtari, Mohamed Ali Feki, and Mahmoud Ghorbel (2004). Integration of home networking in a smart environment dedicated to people with disabilities, Information and Communication Technologies: From Theory to Applications, 2004.Proceedings. 2004 International Conference, April 2004, pp. 125 – 126

[18] Ahmad Rabie and Uwe Handmann (2014), NFC-Based personspecific Assisting System in Home Environment, Proceeding of the 11th World Congress on Intelligent Control and AutomationShenyang, China, July 2014, pp. 5404 – 5409

PLAGIARISM REPORT

ORIGINALITY REPORT

9%	6%	4%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	components101.com Internet Source	<1%
2	docplayer.net Internet Source	<1%
3	oaf91.revistapasajes.com Internet Source	<1%
4	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1%
5	ejaet.com Internet Source	<1%
6	www.idc.com Internet Source	<1%
7	aws.amazon.com Internet Source	<1%
8	Submitted to Swinburne University of Technology Student Paper	<1%