HASSIBA BENBOUALI DE CHLEF UNIVERSITY

Faculty of Technology

Department of Electronics

MASTER'S THESIS

Field: : SCIENCE AND TECHNOLOGY

Sector: : Electronics

Specialty : Electronics of embedded systems

Theme

Design and Implementation of a Secure Access Control System Using Biometric Authentication.

By

SERAI MAAMAR

GUELAMINE NABIL

Supervisor:

Dr. DJEGHLOUF ASMAA

Assistant Professor B UHBC

Chlef, Juin 2025

HASSIBA BENBOUALI DE CHLEF UNIVERSITY

Faculty of Technology

Department of Electronics

MASTER'S THESIS

Field: : SCIENCE AND TECHNOLOGY

Sector: : Electronics

Specialty : Electronics of embedded systems

Theme

Design and Implementation of a Secure Access Control System Using Biometric Authentication.

By

SERAI MAAMAR

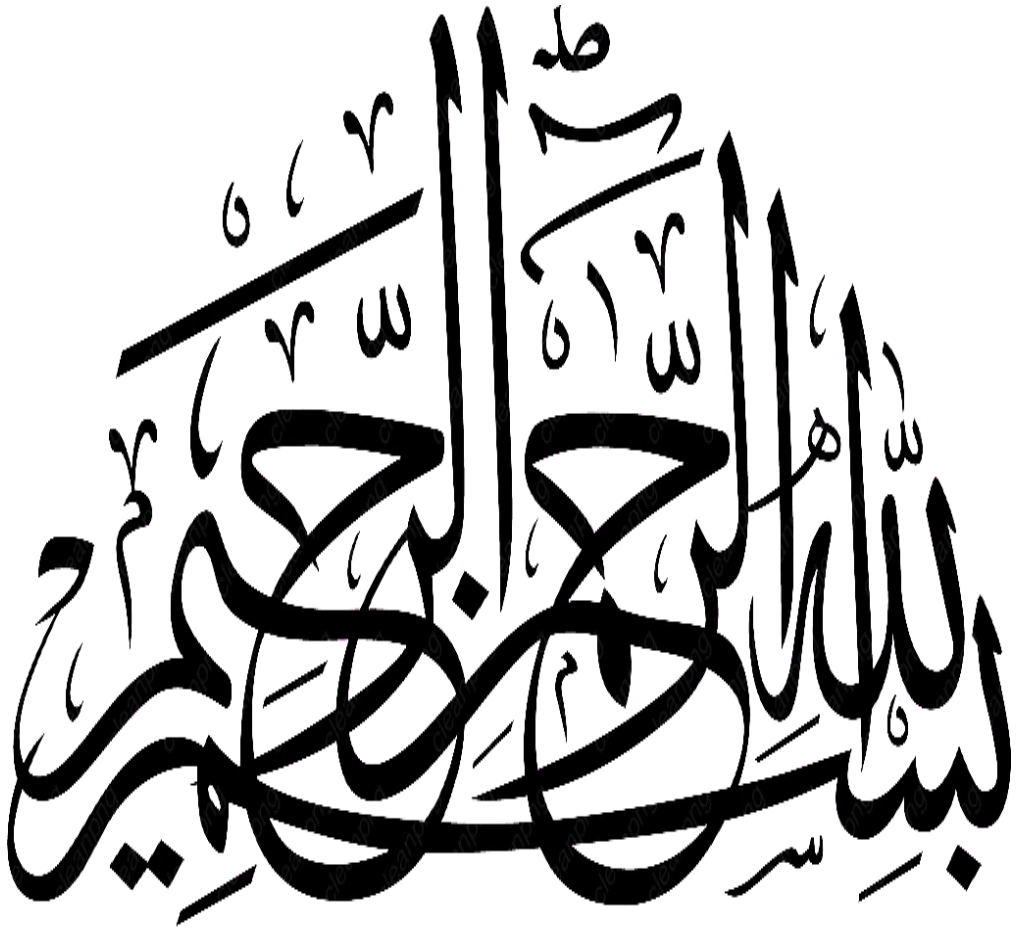
GUELAMINE NABIL

Supervisor:

Dr. DJEGHLOUF ASMA

Assistant Professor B UHBC

Chlef, Juin 2025


بسم الله الرحمن الرحيم

Dedication

بسم الله الرحمن الرحيم

الحمد لله الذي تتم بنعمته الصالحات

والصلاة والسلام على الحبيب المصطفى (صلى الله عليه وسلم)

الى **امي** و**ابي** الحبيبين لقد كنتم لي سندا وعونا في مسيرتي الدراسية ورافقتموني معنويا وماديا اعلم أني لن اوفي حقكم ولو عملت حياتي كاملة في خدمتكم ولكن اهدي تخرجي لكم مقابل ذرة مما قدمتماه ليحفظكم الله وادامكم تاجا فوق رؤوسنا واتمنى ان انال بركم.

اخوتي **عبد الغني، يونس، عبد الوهاب، احمد، لقمان** و**معاذ** و**اختاي العزيزتين** حفظكم الله وفتح ابواب الخير امامكم.

مشرفتي **الدكتورة جغلوف اسماء** بارك الله فيك على مجهوداتك وحرصك علينا ووفقك في حياتك .

شريكي وصديقي **قلامين نبيل**.

اصدقائي.

الى **اساتذتي** في قسم الالكترونيك.

إلى كل من ساعدني، بشكل مباشر أو غير مباشر، في دراستي.

وإلى كل من ساهم في الابتكار وتحسين حياة البشرية.

Maamar

Dedication

بسم الله الرحمن الرحيم

الحمد لله الذي تتم بنعمته الصالحات

والصلاة والسلام على الحبيب المصطفى (صلى الله عليه وسلم)

Avant tout, je tien à remercies le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes de la vie.

Je Dédié ce travail :

À ceux qui me sont chers à ceux qui ont toujours cru en moi à ceux qui m'ont toujours encouragé

À ma très chère mère

La lumière de mes jours, la source de mes efforts, la flamme de Mon cœur, ma vie et mon bonheur, qui m'a entouré d'amour, d'affection et qui fait tout pour ma réussite, maman que j'adore, que dieu la garde.

À mon très cher père

La Source de joie et de bonheur, celui qui s'est toujours Sacrifié pour me voir réussir, qui m'a aidé à devenir ce que je suis aujourd'hui, que dieu le garde et le protège.

À Madame **Asmaa DJEGHLOUF**

notre encadrante, pour sa disponibilité, ses conseils précieux, son accompagnement bienveillant et son soutien tout au long de ce travail. Merci pour votre professionnalisme et votre implication.

À mes chers frères Fethi, Rafik et Abdelhakim.

À mon binôme et meilleure ami Serai Maamar.

À mes enseignants du département de l'électronique.

Et tous les proches amis sans exception.

Nabil

Acknowledgments

First, we thank God for the courage, determination, and patience He has granted us to successfully complete this project.

We extend our sincere thanks to our supervisor, **Dr. DJEGHLOUF Asmaa**, for her patience, cooperation, efforts, and valuable advice, which have been of great help. Despite the distance from the university where she currently works, she did not leave us alone and provided us with objective critiques of our progress. Allow us, Madam, to express our gratitude and respect for you.

Thank you to the jury for agreeing to discuss our master's thesis. We hope you like our work.

We extend our sincere thanks to the head of the department and all the professors in the Electronics Program.

Finally, we extend our deepest gratitude to our families for their presence and moral support.

Abstract

As part of our final project, we developed a biometric access control system based on two-factor authentication. The system utilizes an ESP32-WROOM-32 microcontroller and communicates with the Firebase Realtime Database to store and transmit data. It provides real-time user feedback through Telegram notifications, web interface messages, and LCD display prompts. Access is granted when both the fingerprint and password are verified within the allowed number of attempts; otherwise, access is denied. The system was simulated using Proteus and programmed in C++ via the Arduino IDE.

Future improvements may include the integration of multimodal biometric authentication to enhance security and system reliability.

Keywords: Access Control System, Biometrics, Embedded Systems, ESP32-WROOM-32, Firebase Realtime Database, Fingerprint, Telegram API.

Résumé

Dans le cadre de notre projet de fin d'études, nous avons développé un système de contrôle d'accès biométrique basé sur une authentification à deux facteurs. Le système repose sur un microcontrôleur ESP32-WROOM-32 et utilise la base de données en temps réel Firebase pour stocker et transmettre les données. Il fournit un retour d'information en temps réel à l'utilisateur via des notifications Telegram, des messages affichés sur une page web et un écran LCD. L'accès est accordé si l'empreinte digitale et le mot de passe sont corrects et saisis dans la limite du nombre de tentatives autorisées ; dans le cas contraire, l'accès est refusé. Le système a été simulé à l'aide de Proteus et programmé en C++ via l'environnement de développement Arduino IDE.

Des améliorations futures pourraient inclure l'intégration d'une authentification biométrique multimodale afin de renforcer la sécurité et la fiabilité du système.

Mots-clés : Système de contrôle d'accès, Biométrie, Systèmes embarqués, ESP32-WROOM-32, Firebase Realtime Database, Empreinte digitale, API Telegram.

ملخص

في إطار مشروع نهاية الدراسة، قمنا بتطوير نظام للتحكم في الوصول البيومتري يعتمد على المصادقة الثنائية. يعتمد النظام على المتحكم الدقيق ESP32-WROOM-32 ويستخدم قاعدة بيانات Firebase اللحظية لتخزين البيانات وإرسالها في الوقت الحقيقي. يوفر النظام تغذية راجعة فورية للمستخدم من خلال إشعارات Telegram ، ورسائل تظهر على صفحة ويب، وشاشة .LCD يتم منح الوصول في حال كانت بصمة الإصبع وكلمة المرور صحيحتين وتم إدخالهما في حدود عدد المحاولات المسموح بها، وإلا يتم رفض الوصول. تم محاكاة النظام باستخدام برنامج Proteus وتمت برمجته بلغة C++ من خلال بيئة التطويرArduino IDE.

قد تتضمن التحسينات المستقبلية دمج نظام مصادقة بيومترية متعددة الوسائط من أجل تعزيز أمن وموثوقية النظام.

الكلمات المفتاحية: نظام التحكم في الوصول، القياسات الحيوية، الأنظمة المدمجة، ESP32-WROOM-32، قاعدة بيانات Firebase اللحظية، بصمة الإصبع، واجهةTelegram AP.

List of Figures

List of tables

List of Abbreviations

ADC: Analog-to-Digital Converter

AES: Advanced Encryption Standard

AP: Access Point

API: Application Programming Interface

ARES: Advanced Routing and Editing Software

CCD: Charge-Coupled Device

CMOS: Complementary Metal-Oxide-Semiconductor

CSS: Cascading Style Sheets

DAC: Digital-to-Analog Converter

DNA: Deoxyribonucleic Acid

DSP: Digital Signal Processor

ECC: Elliptic Curve Cryptography

EDA: Electronic Design Automation

En: Enable

FAR: False Acceptance Rate

FRR: False Rejection Rate

GPIO: General Purpose Input/Output

GSM: Global System for Mobile Communications

HTML: Hypertext Markup Language

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

I2C: Inter-Integrated Circuit

ID: Identifier / Identification

IDE: Integrated Development Environment

IEEE: Institute of Electrical and Electronics Engineers

IoT: Internet of Things

IP: Internet Protocol

ISIS: Intelligent Schematic Input System

IT: Information Technology

JSON: JavaScript Object Notation

LAN: Local Area Network

LCD: Liquid Crystal Display

LED: Light Emitting Diode

LEDC: LED Controller

MCU: Microcontroller Unit

MFA: Multi-Factor Authentication

MISO: Master In Slave Out

MOSI: Master Out Slave In

NFC: Near Field Communication

NTP: Network Time Protocol

PCB: Printed Circuit Board

PIN: Personal Identification Number

PWM: Pulse-Width Modulation

RF: Radio Frequency

RFID: Radio-Frequency Identification

RSA: Rivest–Shamir–Adleman

SCK: Serial Clock

SCL: Serial Clock Line

SDA: Serial Data Line

SDKs: Software Development Kits

SHA-2: Secure Hash Algorithm 2

SoC: System on Chip

SPI: Serial Peripheral Interface

SS: Slave Select

SSID: Service Set Identifier

STA: Wi-Fi Station Mode

TTL: Transistor-Transistor Logic

UART: Universal Asynchronous Receiver-Transmitter

UI: User Interface

USB: Universal Serial Bus

VSPI: Virtual SPI

Wi-Fi: Wireless Fidelity

WPA: Wi-Fi Protected Access

Table of contents

Chapter 3: Software Design of the Biometric Access Control System

Chapter 4: Implementation Results and System Validation

General Introduction

General Introduction

Security is a fundamental concern in many fields, whether it involves protecting data, infrastructure, or individuals. With the evolution of technology and the rise of threats, it has become essential to implement robust security mechanisms to prevent unauthorized access and ensure system integrity.

Among the most common security measures, access control systems play a crucial role in restricting entry to specific areas or resources only to authorized individuals. These systems can be based on various authentication methods, ranging from passwords and magnetic cards to more advanced technologies such as biometrics.

Biometrics, which relies on identifying individuals based on their physiological or behavioral characteristics (fingerprints, facial recognition, iris scanning, etc.), offers a reliable and effective solution for access control. Unlike traditional methods, it provides enhanced security by reducing the risks associated with the loss or theft of credentials.

In this context, our final year project aims to design an efficient biometric access control system capable of ensuring fast and secure user identification. This project aligns with the growing need to enhance the security of sensitive areas and strengthen the protection of data and infrastructure against malicious intrusions.

This thesis is divided into four chapters organized as follows:

The first chapter provides a comprehensive introduction to biometric authentication, emphasizing the architecture and functions of biometric access control systems. It explores their practical applications, addresses major challenges such as privacy concerns and accuracy issues, and examines recent technological developments. This overview lays the groundwork for the in-depth analysis of system design and implementation presented in the subsequent chapters.

The second chapter delves into the criteria for selecting hardware components, outlining their specific functions and technical characteristics. It also details how these elements interact within the system and explains the reasoning behind the architectural decisions, ensuring alignment with both functional and security objectives.

The third chapter opens with an overview of the software architecture and the development environment, setting the stage for a closer look at the programming languages employed particularly those used within the Arduino platform. It proceeds to break down the structure of an Arduino sketch, covering aspects such as library imports, variable declarations, and the essential setup () and loop () functions. Special attention is given to the integration of critical software libraries that facilitate fingerprint authentication, wireless connectivity, cloud database access, HTTP communication, and real-time operations.

The final chapter presents the practical implementation and comprehensive validation of the biometric access control system. It begins by detailing the experimental setup, including hardware integration, real-time software behavior, and the flow of user interaction. A simulation using Proteus is conducted to verify the logical functioning of components. The chapter then evaluates system performance based on connectivity, user registration, authentication accuracy, cloud synchronization, and feedback mechanisms. Validation metrics such as FAR, FRR, and system accuracy are analyzed, demonstrating high system reliability under real-world conditions. The web interface is also showcased, allowing real-time monitoring of access events through Firebase.

We will conclude this thesis with a general conclusion in which we summarize the results obtained throughout this work.

Chapter 1:

An overview of Biometric Access Control Systems

Chapter 1: An overview of Biometric Access Control Systems

1.1. Introduction

Biometric authentication has become a key solution for enhancing security in access control systems, offering reliable identification based on unique physiological or behavioral traits such as fingerprints, face, or voice. Unlike traditional methods (passwords, cards), biometrics provide a higher level of security by verifying who you are rather than what you know or have.

This chapter presents an overview of biometric authentication, focusing on the structure and role of biometric access control systems. It highlights their applications, discusses key challenges like privacy and accuracy, and reviews recent technological advancements. This foundational knowledge supports the detailed exploration of system design and implementation in the following chapters.

1.2. Fundamentals of Biometric Authentication

Biometric authentication is a security process that relies on the unique biological characteristics of individuals to verify they are who they say they are.

Biometric authentication systems compare physical or behavioral traits to stored, confirmed, authentic data in a database. If both samples of the biometric data match, authentication is confirmed. Typically, biometric authentication is used to manage access to physical and digital resources, such as buildings and computing devices **[1].**

1.2.1. Definition of biometrics

The term biometrics is derived from the Greek word "**bio**", meaning life, and "**metric**", meaning to measure. Biometrics is the measurement and statistical analysis of people's unique physical and behavioral characteristics. The technology is mainly used for identification and access control or for identifying individuals who are under surveillance. The basic premise of biometric authentication is that every person can be accurately identified by intrinsic physical or behavioral traits **[2].**

1.2.2. Types of biometric traits

Biometric traits are unique biological and behavioral characteristics used to identify or verify individuals. These traits fall into two main categories **(see Figure 1.1)** which is physiological and behavioral biometrics.

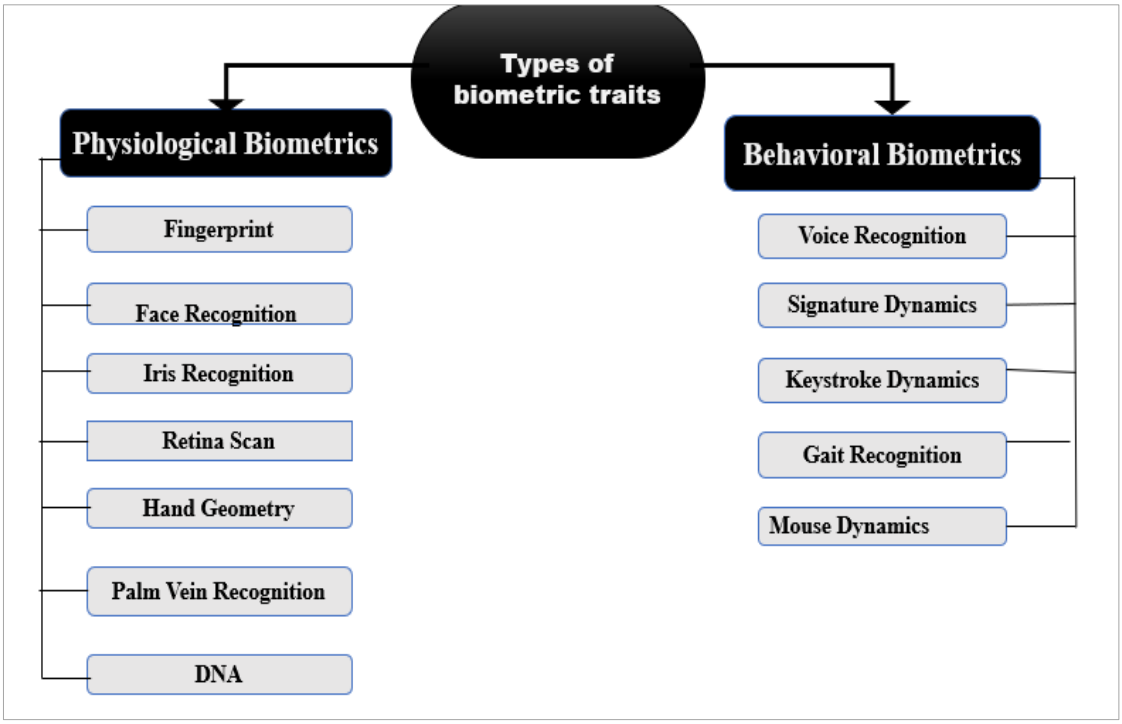


Figure1.1: Categories of biometric traits.

1.2.2.1. Physiological Biometrics

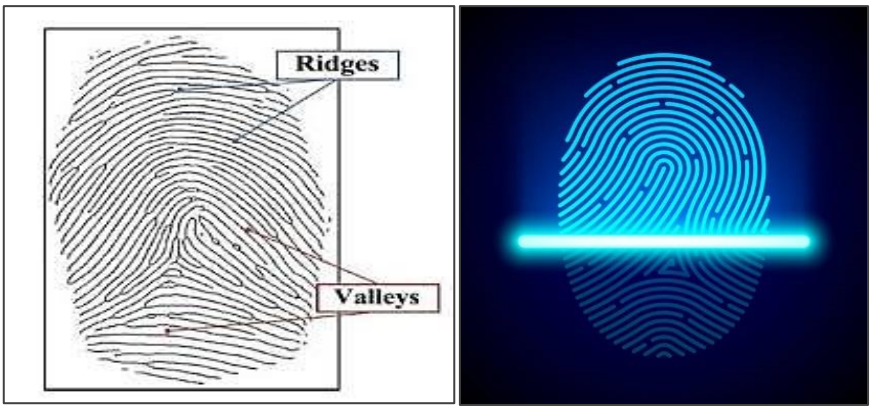
Physiological biometrics are based on **physical characteristics** of an individual that are unique, measurable, and generally stable over time. These traits are **inherent to the human body** and typically do not change significantly throughout a person's life, making them reliable for identification and authentication purposes.

Common examples of physiological biometrics include:

A. Fingerprint

In biometrics science, a fingerprint is the texture pattern formed by the interleaved ridges and valleys on the fingertips (Figure1.2(a)). Fingerprint recognition (Figure1.2(b)) is one of the most popular and successful methods used for person identification, which takes advantage of the fact that the fingerprint has some unique characteristics called minutiae which are the points where the lines of the ridges begin, end, branch off and merge.

Fingerprint is the most dominant modality in the market; it constitutes a trade-off in terms of accuracy, security and cost among other modalities **[3].**



(a) **(b)**

Figure 1.2: (a) Fingerprint Ridges and valleys **(b)** Fingerprint recognition **[4].**

B. Face Recognition

Facial recognition systems (Figure 1.3) are capable of verifying or identifying a person's identity by analyzing their unique facial features from an image. A form of biometrics, facial recognition relies on computer vision and artificial intelligence algorithms to work.

Face recognition software leverages artificial intelligence, image recognition and other advanced technology to map, analyze and confirm the identity of a face. This can then be used to identify specific people in photos and videos, determine if a face in different images belongs to the same person, and search for a particular face among a large collection of images **[5].**

Figure 1.3: Facial recognition **[6].**

C. Iris Recognition

Iris recognition (Figure1.4) is one type of bio-metric method used to identify the people based on single patterns in the region of ring-shaped surrounded the pupil of the eye. Generally, the iris has a blue, brown, gray or green color with difficult patterns that are noticeable upon close inspection **[7].**

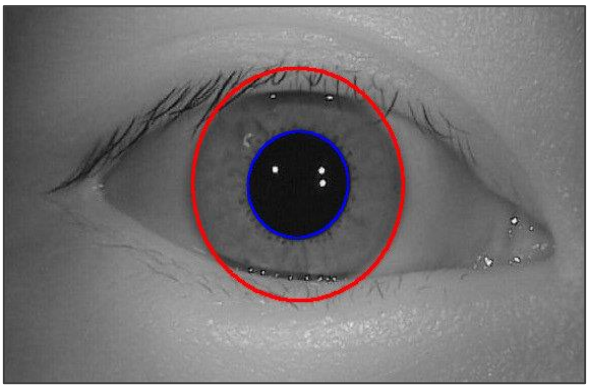


Figure 1.4: Iris recognition **[8].**

D. Retina Scan

Retina scanning **(Figure1.5)** is a biometric authentication technology that uses an image of an individual's retinal blood vessel pattern as a unique identifying trait for access to secure installations, a retina scan requires the subject to focus on a single point for 15 seconds **[9].**

Figure 1.5: Retina scanning **[10].**

E. Hand Geometry

Hand geometry biometric systems (Figure1.6) incorporate the salient features of finger geometry, but also include the surfaces of the hand itself and its side profile. Images are taken while the hand is kept palm down on a support plate and kept in position by the use of guide pegs. The length, width, thickness, and surface area of the individual's hand is measured and recorded. Multiple features and measurements are extracted during this process. Several images of the same hand are often taken in order to produce a single stored template that has sufficient detail for identification purposes. These images and the supporting data are then filed in a database and are used to authenticate the identity of the enrollee in subsequent encounters when the subject's hands are imaged again and compared with the reference images to confirm or reject the identity claim **[11].**

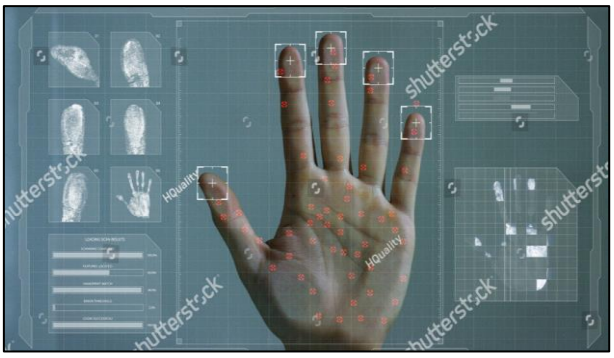


Figure 1.6: Hand geometry **[12].**

F. Palm Vein Recognition

Palm vein scanning (Figure1.7), an advanced biometric technique, captures and analyses the intricate patterns of veins in an individual's palm. Uniquely encoded by nature, these patterns serve as a highly accurate and reliable means for identity verification.

Operating on the fascinating principle that haemoglobin in the blood absorbs infrared light, the technology unveils distinct vein patterns beneath the skin's surface. These hidden vascular structures become the blueprint for individual identification, forming the cornerstone of palm vein scanning's precision **[13].**

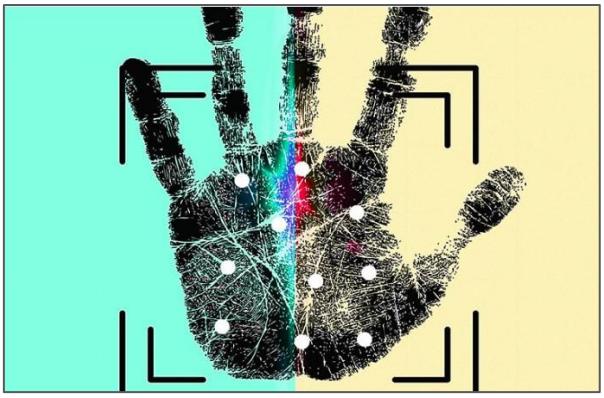


Figure 1.7: Palm vein scanning **[13].**

G. Deoxyribonucleic Acid (DNA)

DNA biometrics (Figure1.8) is a very reliable technique that uses genetic fingerprinting. This fingerprint is obtained following an analysis of biological tissues such as hair, blood, and saliva. Genetic fingerprint recognition is one of the most secure and accurate technologies. However, DNA analysis cannot, for the moment, be adapted to rapid recognition and is expensive, since it requires specific analysis laboratories. Therefore, its use is limited to the recognition of family ties or criminals. DNA is not widely used for logical and physical access control. **[14]**.

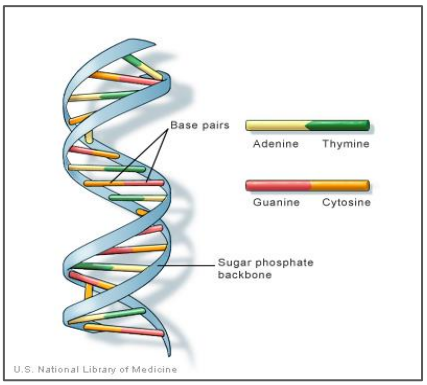


Figure 1.8: Deoxyribonucleic acid (DNA) **[15].**

1.2.2.2. Behavioral Biometrics

Behavioral biometrics analyze the way a person interacts with their mobile device, such as how they hold their phone or how fast they type. As a result, behavioral biometrics create a pattern of behavior that is unique to a person.

Common examples of behavioral biometrics include:

A. Voice Recognition

Voice recognition (Figure 1.9) is a highly desirable feature in remote application systems, such as authenticating a person over the phone. However, its use is highly sensitive due to its high sensitivity to external conditions (such as illness, the person's stress, etc.). It is sometimes chosen in conjunction with other features, such as voice and typing [**16].**

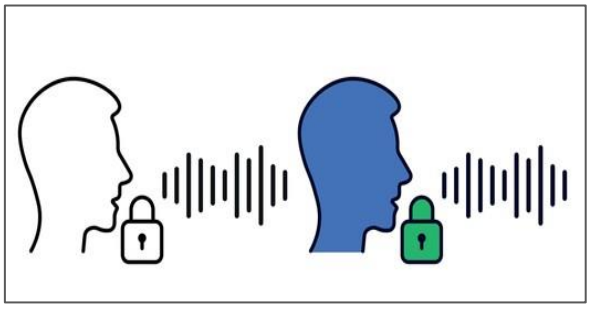
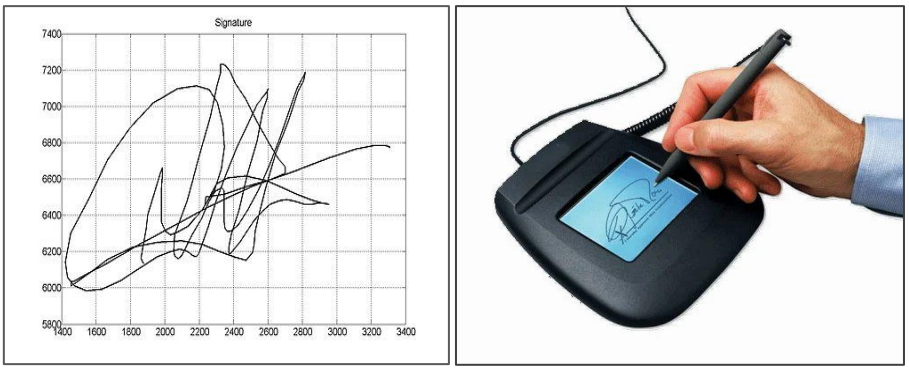


Figure 1.9: Voice recognition **[17].**

B. Signature Dynamics

Signature recognition involves measuring several factors: speed, movement, pen pressure, acceleration, etc. The device is usually attached to a graphics tablet with a pen. The advantage of this method is that it is widely accepted by the public. However, its drawback is the reproducibility of the signature by the same person. Furthermore, the nature of the signature depends on several factors, such as stress, age, and fatigue, which hinder recognition **[18].**



(a) **(b)**

Figure 1.10: **(a)** Example of signature shape (b) Signature recognition **[19].**

C. Keystroke Dynamics

Keystroke Dynamics (Figure1.11**),** also known as typing biometrics or keyboard dynamics, refers to the study and analysis of the unique patterns in the way an individual types on a keyboard. It captures the subtle nuances in typing speed, rhythm, pressure, and the time intervals between keystrokes. By creating a user profile based on these characteristics, Keystroke Dynamics serves as a biometric authentication method **[20].**

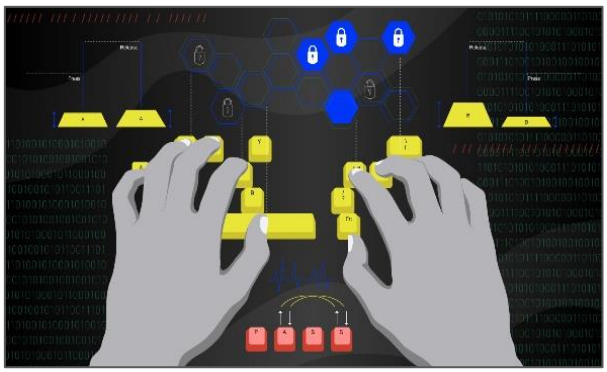


Figure 1.11: Keystroke Dynamics **[21].**

D. Gait Recognition

Gait refers to the manner in which a person walks (Figure1.12), and is one of the few biometric traits that can be used to recognize people at a distance. Therefore, this trait is very appropriate in surveillance scenarios. Most gait recognition algorithms attempt to extract the human silhouette in order to derive the gait variables. Hence, the selection of a good model to represent the human body is pivotal to the efficient functioning of a gait recognition system **[22].**

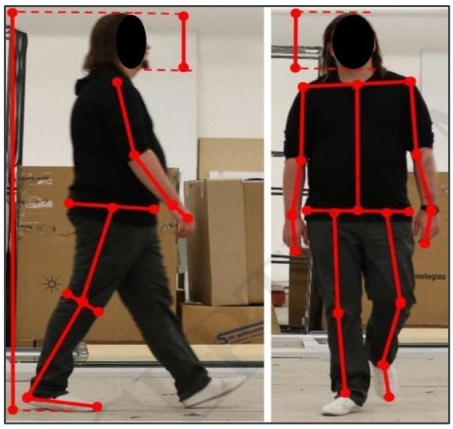


Figure 1.12: Gait recognition **[23].**

E. Mouse Dynamics

Mouse Dynamics (Figure 1.13) is a behavioral biometrics technology used to validate a user's identity by analyzing unique patterns (such as tiny hand motions) detected in the user's interaction with their mouse or pointer. Because it enables continuous authentication, mouse dynamics is a great fit for intrusion detection solutions **[24]**.

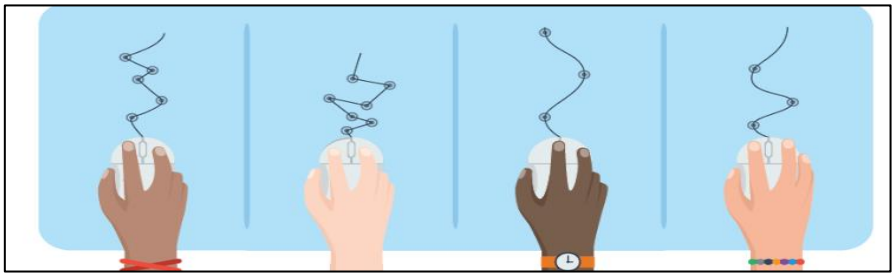


Figure 1.13: Mouse Dynamics **[25].**

1.3. Biometric Access Control Systems

A biometric access control system can be used to enhance building or facility security systems by adding an additional layer of verification. Unlike traditional access control systems that use access cards or tokens, a biometric access control system utilizes a person's physical characteristics, such as fingerprints, facial features, palm veins, and iris. These characteristics cannot be copied, improving the accuracy of identification and authentication. Additionally, a biometric access control system also offers the option of touchless access control **[26].**

Access control systems related to biometric authentication use biometric data to control access to physical spaces or digital systems. These systems typically use one or more biometric traits, such as fingerprints, facial recognition, iris scans, or voice recognition, to verify the identity of a user and grant or deny access accordingly.

In a physical access control system, biometric authentication can be used to control access to buildings, rooms, or secure areas. For example, employees may use their fingerprints or facial recognition to gain access to a restricted area in a building, or visitors may need to scan their iris to enter a secure facility.

In a digital access control system, biometric authentication can be used to control access to computer systems, networks, or applications. For example, a user may need to scan their fingerprint or speak a passphrase to access their computer or log into a secure website **[27].**

1.3.1. Components of a Biometric Access Control System

The Figure1.14 represents the Illustration of the biometric system workflow.

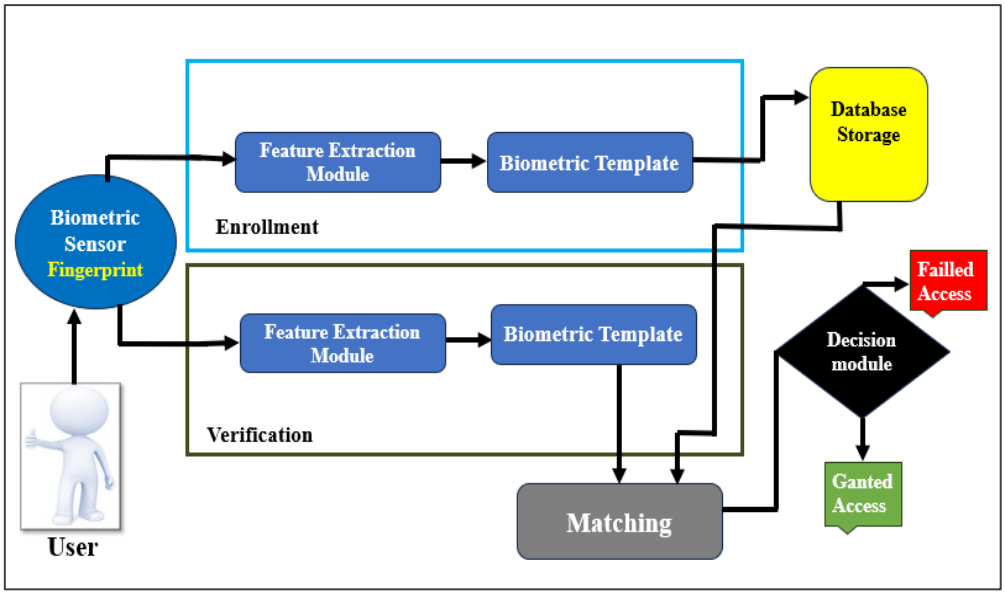


Figure 1.14: Illustration of the Biometric System Workflow.

The Biometric access control System include:

1.3.1.1. Biometric Sensor (Data Acquisition Module)

The biometric sensor (fingerprint, camera...) is a mediator between the system and the real world. It collects all the required data from the users. It is an image acquisition technique, retrieving the image details from external sources to perform further processes **[28].**

1.3.1.2. Feature Extraction Module

Feature extraction is a technique that converts the image raw data into numerical data. Feature extraction techniques perform various operations to remove the redundancy from the image. After removing the redundancies from the input image, some unique features of the scanned image get stored in the database. All the extracted features from the image are

combined to generate the template. These templates are in the form of vectors or numerical data **[28].**

1.3.1.3. Database Storage

It maintains a record of the number of users. It stores their biometric scan and helps the matcher in the process of identifying the user **[28].**

1.3.1.4. Matcher (Matching Module)

In this process, whenever a person wants access control of any file or property, the matcher first enquires the database whether they have such a user or not by extracting the feature from that scanned image. Matcher compares it with the various data of the scanned images **[28].**

1.3.1.5. Decision Module

If the person's scanned images match with the stored data, the biometric system permits the person to get in. In case the person does not have any record in the database, it will flash a message as NO RECORDS FOUND or something like that. The output generation depends upon the pre-processing of the input **[28].**

1.3.1.6. User Interface

Biometric access control provides feedback with users like green light if the access is authorized and send denied message if the access refused

1.3.2. Biometric Access Control System Vs traditional authentication methods

When comparing biometrics vs. traditional authentication methods like password (Figure1.15), biometrics are more secure. Like Passwords can be shared, stolen, or hacked, compromising security. In contrast, biometrics eliminates these risks using physical characteristics like fingerprints or facial recognition that cannot be easily duplicated.

Furthermore, biometric authentication eliminates the need to remember or use strong passwords, streamlining the security process and reducing the risk of breaches. Passwords and biometrics are used together in two-factor authentication systems, enhancing overall security by requiring both a password and a biometric trait **[29].**

Figure 1.15: Password authentication **[30]**.

1.4. Applications of Biometric Access Control

Biometric access control is widely applied across sectors (Figure1.16) to boost security, convenience, and efficiency. Governments and border agencies use it for immigration and national security, while companies enhance workplace safety and employee management. In healthcare, it safeguards patient data and restricts lab access. Banks and financial services use biometrics for secure authentication and fraud prevention. Smartphones and consumer electronics rely on biometric features for device security and payments. Schools and research institutions manage access to facilities, while smart homes and IoT devices offer personalized, secure control. Law enforcement uses biometrics in criminal investigations, and airports and transportation hubs streamline passenger verification and security processes with biometric systems **[31]**.

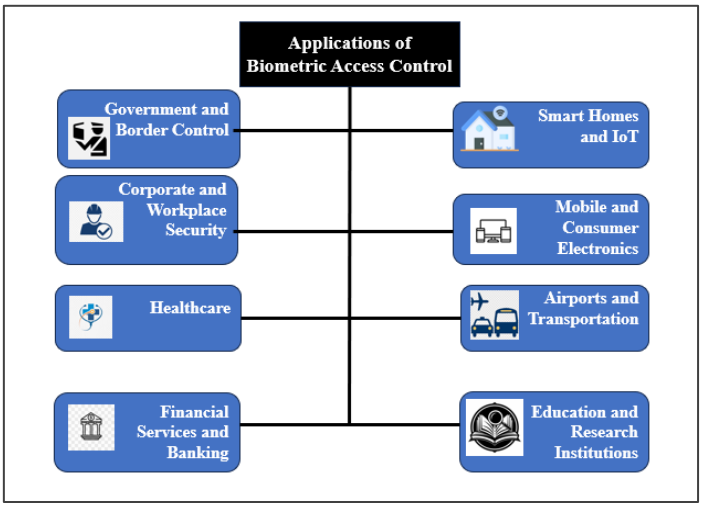


Figure 1.16: Applications of Biometric Access Control.

1.5. Key Challenges in Biometric Access Control

Key challenges in biometric access control span technical, security, privacy, operational, and user acceptance domains. Below is a detailed overview of these challenges and potential ways to address them

1.5.1. Accuracy and Reliability Issues

In order to understand how to determine the performance of a biometric system, we need to define two main criteria:

1.5.1.1. False Acceptance Rate (FAR)

A False Acceptance in **biometric access control** refers to cases where the system grants access to an unauthorised person. Even though that person is not registered in the system, they presented their biometric data and it was accepted. The **False Acceptance Rate** is usually shown as a percentage. It shows the proportion of all unauthorised attempts to enter that result in a False Acceptance **[32].**

The equation **(1.1)** gives the false acceptance rate:

$$FAR(\%) = \frac{number\ of\ accepted\ imposter}{total\ number\ of\ impster\ access} \quad \textbf{(1. 1)}$$

1.5.1.2. False Rejection Rate (FRR)

A False Rejection in biometric access control occurs when an authorised person is not recognised as such by the system. Even though they are correctly registered, the system rejects them and does not allow them access to the secure area. The **False Rejection Rate** is the percentage of legitimate attempts to gain access that are incorrectly rejected by the system **[32].**

The equation **(1.2)** gives the false Rejection rate:

$$FRR(\%) = \frac{Number\ of\ False\ rejection}{total\ number\ of\ genuine\ access} \quad \textbf{(1. 2)}$$

1.5.2. Spoofing and Presentation Attacks

Biometrics can enhance identity management but is not foolproof against fraud or identity theft. Vulnerabilities exist, including spoofing, where fake biometric artifacts deceive sensors. Since computer vision differs from human vision, some spoofing (Figure1.17) methods can be unexpected. Liveness detection helps counter spoofing by distinguishing real users from fake representations, but biometric systems may still be vulnerable to adversarial attacks **[33].**

Figure 1.17: Example of Spoofing attack **[34].**

1.5.3. Privacy, Data Protection and Database Attacks

As an individual, there are several steps you can take to protect your biometric data from being hacked **[35].**

First, be aware of the risks associated with biometric data. When this information is compromised, it can be used to gain access to your accounts and sensitive personal information.

Second, only use reputable biometric devices and services. Be sure to do your research to find out if a company is reputable before you use its products or services.

Third, keep your biometric data secure. Store it in a safe place where only you can access it. Do not share it with anyone else, and be sure to destroy any copies of it that you no longer need.

Fourth, regularly update your biometric data. This will help to ensure that your information is kept up-to-date and secure.

1.5.4. Cost and Maintenance

The cost of biometric systems can be expensive and complex to implement, especially for larger organizations or those with high-security needs. The initial outlay for sophisticated biometric technologies and the ongoing costs associated with maintenance and updates can be significant **[36].**

1.6. Recent Trends and Technological Advancements

Recent trends and technological advancements in biometric access control focus on enhancing security, convenience, privacy, and integration with broader smart systems. Key developments include:

1.6.1. Multimodal biometric systems

Multimodal biometric (Figure1.18) refers to a system that uses multiple personal traits such as iris, fingerprints, face, retina, hand geometry, voice or signatures for identification and verification, providing more accurate and reliable results compared to systems that use only one trait. These systems require fusion frameworks and efficient recognition algorithms, and are widely used in various applications including airport security, access control, suspect identification, and network security **[37].**

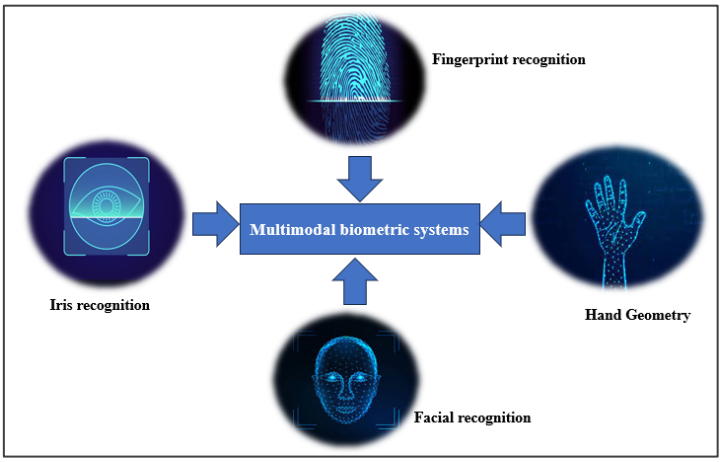


Figure 1.18: Multimodal biometric**.**

1.6.2. Contactless Biometrics

The demand for hygienic, frictionless access control solutions is accelerating as organizations prioritize health, safety, and convenience because Accelerated by the **COVID-19** pandemic and hygiene concerns. Touchless systems (Figure1.19), including mobile credentials and biometric technologies like facial recognition, are leading the charge. For instance, mobile credentials stored on smartphones or wearable devices allow users to access secure areas using contactless technologies **[38].**

Figure 1.19: Touchless systems **[39].**

1.6.3. Edge computing and embedded systems in biometric authentication

Edge computing (Figure1.20) is defined as the storage and processing of data by resources deployed in closer proximity to the end-user or data collection point, rather than a distant data center or cloud. This proximity can be used in latency-reducing architectures, and could also have privacy, regulatory, and cost benefits. Many predicted growth areas for biometrics and edge computing overlap, including smart cities and the internet of things (IoT) **[40].**

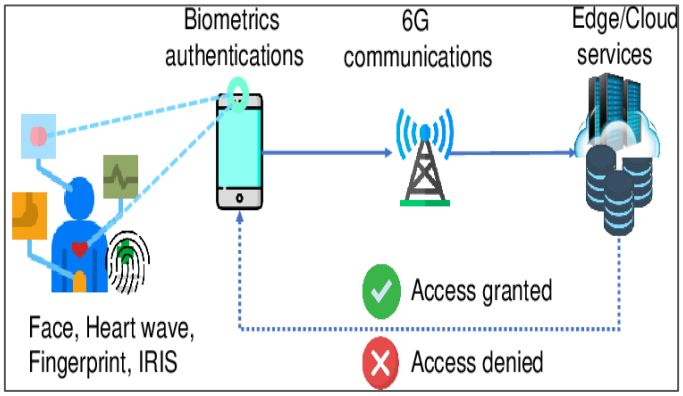


Figure 1.20: Edge computing in biometric **[41].**

1.6.4. Integration with Mobile and Wearable Devices

Mobile credentials are rapidly replacing traditional physical cards as businesses embrace touchless and mobile or smartwatch -first access solutions (Figure1.21). Near Field Communication (NFC)-enabled credentials, stored in digital wallets, provide a secure and convenient way for users to access facilities with a simple tap to a reader. NFC is a short-range wireless communication technology that enables secure data exchange between devices in close proximity, typically within a few centimeters. This close-range interaction makes NFC inherently more secure, as it reduces the likelihood of unauthorized interception or eavesdropping **[40].**

Figure 1.21: Fingerprints biometric payment **[42].**

1.6.5. Cloud-Based Biometric Services

Cloud-based biometric (Figure1.22) solutions authenticate and grant access to users by using an image or template of uniquely human traits such as fingerprints, facial recognition, iris recognition, and others. At registration, a user must enter a single or many biometric features that will be saved as templates on the cloud's server. The most impressive feature of cloud-based biometrics is that it encrypts the fingerprint templates and images provided by the user each time they want to verify their identity **[43].**

Figure 1.22: Cloud-based biometric **[44].**

1.7. Conclusions

In this chapter, we have introduced Biometric Authentication, we specifically focus on the Biometric Access Control system with types(physiological/behavioral), components, how to works and some applications.

Finally, we describe Key Challenges in Biometric Access Control and Recent Trends and Technological Advancements.

Chapter2:

Hardware Design of the Biometric Access Control System

Chapter 2: Hardware Design of the Biometric Access Control System

2.1. Introduction

The reliability and effectiveness of any biometric access control system heavily depend on the design and integration of its underlying hardware components. This chapter presents a comprehensive overview of the hardware architecture implemented for the secure fingerprint-based access control system. The design focuses on selecting and interfacing components that ensure accurate biometric acquisition, efficient processing, secure communication, and robust actuation.

At the core of the system lies the ESP32 microcontroller, chosen for its computational capabilities, integrated wireless communication modules, and compatibility with biometric peripherals. A fingerprint sensor module is employed as the primary biometric acquisition device, enabling the system to uniquely identify individuals based on their physiological traits. Additional components such as the power supply unit, relay module for door control, and peripheral interfaces are discussed in detail.

This chapter elaborates on the selection criteria, functional roles, and technical specifications of each hardware element. It also explains the interconnections between components and highlights the rationale behind the architectural choices made to meet the system's functional and security requirements.

2.2. Proposed System Overview

The proposed biometric access control system presented in Figure2.1 is designed to enhance the security of physical spaces by using fingerprint recognition as a means of authenticating users. The system architecture integrates multiple hardware components, each fulfilling a specific role in the access control process. At the heart of the system is the ESP32 microcontroller, which coordinates all operations, including biometric data acquisition, user interface management, decision-making, and actuation.

When a user attempts to gain access, the system prompts them to place their finger on the fingerprint sensor. The sensor captures the fingerprint image and either performs the matching internally or transmits the data to the ESP32 for further processing. Upon successful authentication, the system activates a relay module to unlock the door, while simultaneously providing visual and auditory feedback using LEDs and a buzzer.

User interaction is further facilitated through a 4x3 keypad, which can be used for additional input such as PIN codes, user IDs, or menu navigation. System messages and status information are displayed on an LCD screen, allowing users to receive real-time feedback and instructions during the authentication process.

To ensure reliability and user-friendliness, the system is designed to operate autonomously and in real-time, with a focus on fast response and low power consumption. The ESP32's built-in Wi-Fi and Bluetooth capabilities also make it possible to implement optional remote monitoring or integration with cloud-based management platforms in future enhancements.

This modular and flexible architecture allows for scalability, where additional biometric sensors or access control points can be integrated as needed, without significant modification to the core system.

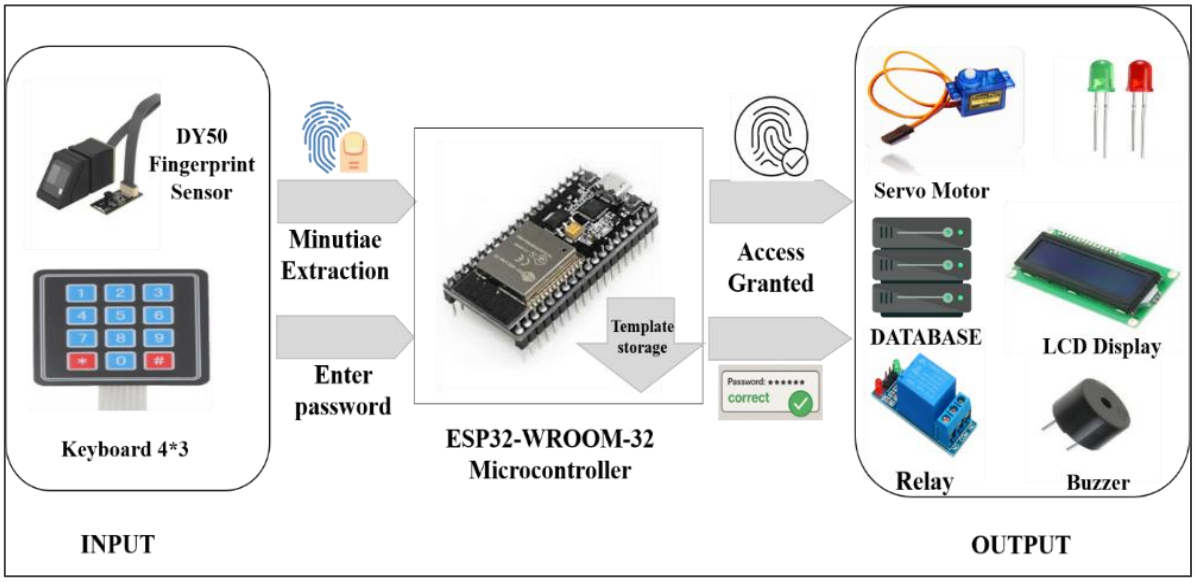


Figure 2.1: Biometric access system overview.

As shown in Figure 2.1, the system is divided into three main parts: the input devices (DY50 fingerprint sensor and 4×3 keypad), the microcontroller (ESP32-WROOM-32), and the output devices (LEDs, buzzer, LCD display, etc.). A detailed description of each component is provided in the following sections.

2.3. Fingerprint Sensor Module

The fingerprint sensor is a critical component in biometric authentication systems and exists in several technological variants, including:

- Optical Fingerprint Sensor
- Capacitive Fingerprint Sensor

- Ultrasonic Fingerprint Sensor
- Thermal Fingerprint Sensor **[45].**

The DY50 optical fingerprint sensor (Figure 2.2), manufactured by Adafruit, is a compact and reliable biometric device designed to authenticate users based on the unique patterns of their fingerprints. It is widely utilized in security systems, access control mechanisms, and other biometric applications. The module features a UART serial interface, allowing for easy integration with microcontrollers such as the ESP32, making it a popular choice for both professional and educational projects **[46].**

Figure 2.2: Optical fingerprint sensor DY50 **[47].**

2.3.1. Working Principle of DY50 sensor

The DY50 optical fingerprint sensor is engineered with multiple integrated components that collectively enable accurate fingerprint recognition (Figure 2.3)

At the top of the sensor is a protective glass layer, which shields the internal components from dust, moisture, and other environmental contaminants, thereby enhancing the device's durability and operational reliability.

Beneath this layer, LEDs illuminate the fingertip, ensuring sufficient contrast for image capture. The core component of the sensor is CCD or CMOS image sensor, which captures the ridge and valley patterns of the fingerprint. These patterns are then converted into binary data, with ridges and valleys interpreted as digital “on” or “off” states to form a fingerprint image template.

The sensor incorporates an internal microcontroller unit (MCU) or digital signal processor (DSP) that handles system control, analog-to-digital conversion, and preprocessing tasks such as image enhancement and template extraction. Data is then transmitted to the host microcontroller (e.g., ESP32) through a UART interface, facilitating real-time communication with external devices such as microcontrollers, smartphones, or computers **[45].**

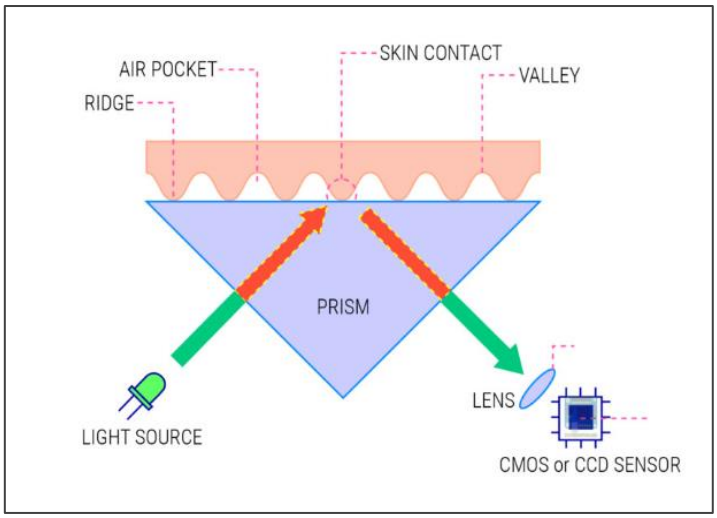


Figure 2.3: Optical fingerprint DY50 principal work. [**48].**

Figure 2.4 illustrates the pin configuration of the DY50 fingerprint sensor. The sensor has four primary pins:

- VCC : 3.3V power supply
- TX : Transmit pin (data output from sensor)
- RX : Receive pin (data input to sensor)
- GND : Ground (power and signal ground) **[49]**

These pins enable UART communication and power connectivity with the microcontroller, typically the ESP32 in this system. Proper connection and voltage matching are essential for reliable operation and to prevent hardware damage.

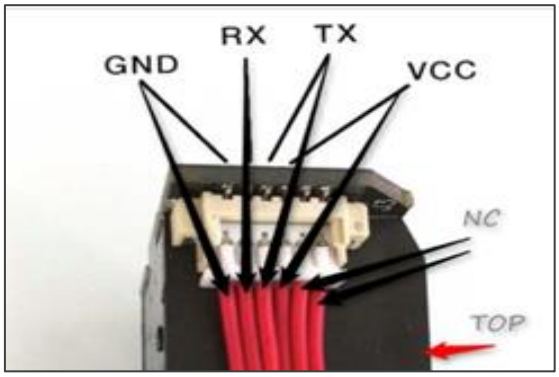


Figure 2.4: Pinout of the DY50 Fingerprint Sensor **[49].**

2.3.2. Enrolment and Matching

Fingerprint processing consists of two main stages: enrollment and matching.

During the enrollment phase, the user is required to scan the same finger twice. The system captures both fingerprint images, processes them, and generates a fingerprint template based on

the extracted features. This template is then stored in the internal database (fingerprint library) for future reference.

In the matching phase, the user places their finger on the optical sensor, which captures the current fingerprint and generates a new template. This template is compared against those stored in the system's fingerprint library to determine identity or verify access.

There are two primary types of fingerprint matching:

- **1:1 Matching (Verification):** The system compares the live fingerprint with a specific pre-stored template to verify the claimed identity of the user.
- **1:N Matching (Identification):** The system searches the entire fingerprint library to find a match for the live fingerprint without prior knowledge of the user's identity.

In both cases, the system returns a result indicating whether the match was successful or failed **[50].**

The location of the matching process depends on the design of the fingerprint system. In modern secure systems, fingerprint matching is often performed locally on the sensor module itself, providing higher security and lower communication latency. In older or more basic systems, matching is performed externally by the host microcontroller, which receives raw or partially processed fingerprint data for comparison **[51].**

2.3.3. Security Considerations

Ensuring the security of a biometric access control system is critical to maintaining user trust and system integrity. One of the primary measures is the use of strong encryption and secure communication protocols. For instance, implementing protocols such as the Secure Device Connection Protocol helps safeguard biometric data during transmission between the fingerprint sensor and the microcontroller, preventing potential interception or tampering.

Another essential strategy is the incorporation of multi-factor authentication (MFA). By requiring users to provide an additional form of authentication, such as a PIN code or RFID card, alongside fingerprint verification, the system introduces an extra layer of security that significantly reduces the risk of unauthorized access.

Regular firmware and software updates are also vital. These updates address known vulnerabilities, improve system performance, and ensure compatibility with the latest security standards. Keeping the system up to date minimizes the risk of exploitation by attackers leveraging outdated components.

To prevent spoofing and presentation attacks, the system can employ advanced liveness detection techniques. These techniques enable the sensor to differentiate between a real, live finger and an artificial replica made from materials such as silicone or printed images, thereby strengthening the system's resistance to fraudulent attempts.

Finally, ongoing auditing and monitoring of the biometric system are necessary for early detection of irregular or suspicious activity. By maintaining logs and employing real-time monitoring tools, administrators can quickly identify and respond to potential security breaches, thereby preserving the overall integrity and reliability of the system.

2.4. Microcontroller Unit– ESP32

The ESP32-WROOM-32 module (Figure 2.5), developed by Espressif Systems, is a low-cost, low-power system-on-chip (SoC) that integrates a powerful processing unit with versatile communication capabilities. It is built around the ESP32-D0WDQ6 SoC and is widely recognized for its integrated Wi-Fi, Bluetooth Classic, and Bluetooth Low Energy (BLE) functionalities, which make it highly suitable for modern IoT and embedded applications **[53].**

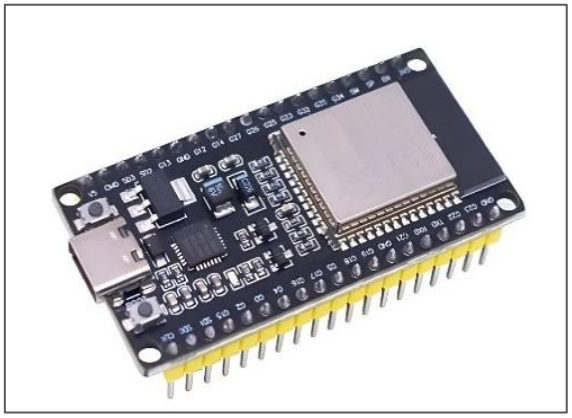


Figure 2.5: ESP32-WROOM-32 Module [**53].**

2.4.1. Selection Criteria for the ESP32 in Biometric Systems

The ESP32-WROOM-32 has been selected for this biometric access control project due to its exceptional combination of processing performance, connectivity options, I/O flexibility, and energy efficiency, all critical attributes for embedded biometric systems.

At the core of the ESP32 is a Tensilica Xtensa LX6 dual-core 32-bit microprocessor, capable of running at speeds up to 240 MHz. This processing power enables the module to efficiently handle demanding tasks such as fingerprint image processing, real-time decision-making, and multitasking, which are essential in responsive and secure biometric applications.

The ESP32 offers a wide range of input/output capabilities, including up to 38 programmable GPIO (General Purpose Input/Output) pins, multiple analog-to-digital (ADC) and digital-to-analog (DAC) channels, pulse-width modulation (PWM), and support for serial communication protocols like UART, SPI, and I2C. These features provide the flexibility to interface with various components, such as fingerprint sensors, LCD displays, keypads, and actuators.

Another key strength of the ESP32 is its power efficiency. It supports several low-power operation modes, with current consumption as low as 5–10 µA in deep sleep mode. This makes it ideal for battery-operated systems and long-term deployments that require minimal maintenance, which is often the case in standalone access control installations.

Beyond hardware capabilities, the ESP32 is widely adopted in biometric systems, including fingerprint and facial recognition technologies. Its real-time performance and built-in wireless capabilities (Wi-Fi and Bluetooth) allow biometric systems to remotely log access attempts, synchronize data, and integrate with cloud platforms or web-based monitoring dashboards. This ensures seamless system management and scalable deployment.

Security is further strengthened by the ESP32's support for encrypted communication protocols, which protect sensitive biometric data during transmission. Moreover, the module is backed by an extensive ecosystem of libraries, documentation, and community support, making it highly practical for rapid development, debugging, and system scaling across a wide range of biometric and IoT applications **[57].**

2.4.2. ESP32 Pinout

The ESP32-WROOM-32 module features a total of 38 pins (Figure 2.6), offering a diverse and rich set of functionalities. These pins can be categorized according to their functions to facilitate clearer understanding and effective utilization in embedded system designs.

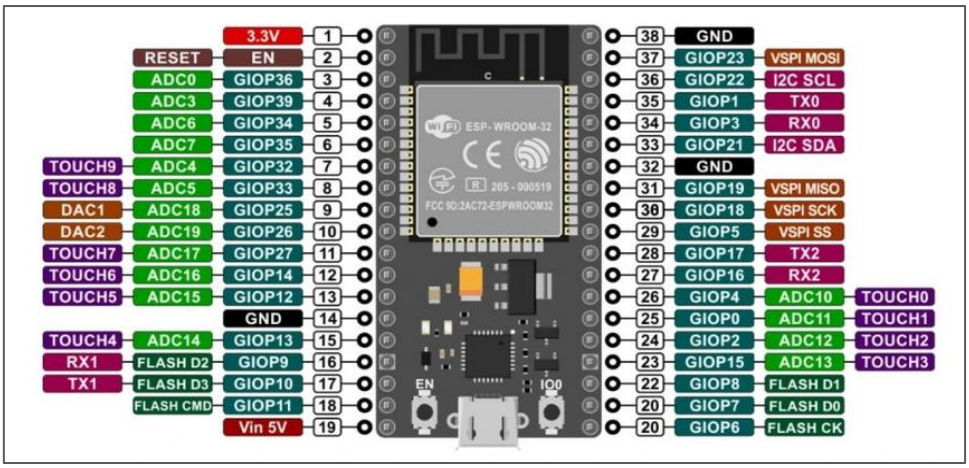


Figure 2.6: ESP32-WROOM-32 Pinout **[54].**

Based on the official Espressif datasheets **[55],** the general-purpose input/output (GPIO) pins can be classified as follows: GPIOs 0, 2, 4, 5, 12–19, 21–23, 25–27, and 32–33 are safe for general digital I/O use. GPIOs 6–11 are internally connected to the SPI flash memory and should be avoided in custom applications. GPIOs 34–39 are input-only pins without internal pull-up or pull-down resistors. Special attention must be paid to GPIOs 0, 2, and 12, which serve bootstrapping functions during the ESP32's startup sequence.

The module also includes analog interfaces, with ADC1 channels available on GPIOs 32–39 and ADC2 on GPIOs 0, 2, 4, 12–15, and 25–27. However, ADC2 channels are restricted when Wi-Fi is in use. Two DAC channels are available on GPIO25 (DAC1) and GPIO26 (DAC2).

In terms of serial communication, the ESP32 supports multiple UART interfaces. UART0, the default interface, uses GPIO1 (TX) and GPIO3 (RX). Additional UART interfaces (UART1 and UART2) can be mapped to various pins, such as GPIO9/10 or GPIO16/17, using the internal GPIO matrix. For SPI communication, up to four interfaces are supported. A commonly used configuration for VSPI is: GPIO23 (MOSI), GPIO19 (MISO), GPIO18 (SCK), and GPIO5 (SS). I2C functionality can also be mapped flexibly, but the default pins are typically GPIO21 (SDA) and GPIO22 (SCL).

The ESP32 supports PWM output on nearly all GPIOs (excluding input-only ones) via its LEDC module, offering up to 16 independent PWM channels. Additionally, capacitive touch sensing is supported on GPIOs 0, 2, 4, 12–15, 27, 32, and 33, enabling proximity-based user interaction.

Key bootstrapping pins include GPIO0 (must be LOW to enter UART bootloader mode), GPIO2 (LOW at boot for proper mode selection), and GPIO12 (controls flash voltage, must be LOW). Power-related pins include 3V3 (regulated 3.3V output), VIN (external 5V input), GND (ground), and EN (enable pin, pulled HIGH to activate the module).

For communication between the DY50 fingerprint sensor and the ESP32 microcontroller, a UART serial interface is typically employed. The sensor’s TX pin is connected to GPIO16 of the ESP32 (configured as RX), and the RX pin of the sensor is connected to GPIO17 (configured as TX). Power is supplied to the sensor through either the 3.3V or 5V pin, depending on the sensor's voltage specifications, while the GND pin of the sensor is connected to the GND pin of the ESP32.

This configuration ensures reliable bidirectional communication between the microcontroller and the biometric sensor, allowing for data exchange during the fingerprint enrollment and matching processes.

2.4.3. Security Features

The ESP32 microcontroller integrates several robust security features that make it well-suited for use in secure access control systems. One of its key advantages is the inclusion of hardware-accelerated encryption modules, which support a range of cryptographic algorithms such as AES (Advanced Encryption Standard), SHA-2 (Secure Hash Algorithm 2), RSA, and Elliptic Curve Cryptography (ECC). These capabilities enable secure data transmission and storage while maintaining high processing efficiency.

Another important feature is Secure Boot, which ensures that only authenticated and signed firmware can be executed on the device. This mechanism protects the system from unauthorized or malicious code execution during startup, thereby maintaining the integrity of the device's firmware.

The ESP32 also offers flash encryption, which encrypts the data stored in its onboard flash memory. This provides an added layer of protection for sensitive information, especially in cases where an attacker gains physical access to the hardware.

In terms of network security, the ESP32 fully supports standard Wi-Fi security protocols, including WPA, WPA2, and the more advanced WPA3, ensuring encrypted and authenticated wireless communication. These features collectively contribute to a secure operational environment for biometric access control systems and other IoT applications **[56].**

2.4.4. Operational Responsibilities of the ESP32

The ESP32 plays a central role in the operation of the biometric access control system, acting as the primary controller responsible for coordinating all key functions. First, it reads and processes data from the fingerprint sensor, either performing the matching internally or relaying data for further processing. Based on the verification result, the ESP32 makes access decisions, either autonomously or in communication with an external server.

Upon successful authentication, the ESP32 triggers the actuation mechanism, such as a relay or servo motor, to unlock the door. It also logs access attempts, which can be stored locally or transmitted wirelessly for remote monitoring. In addition to these core tasks, the ESP32 is

responsible for enforcing security measures, including encrypted data transmission, secure boot verification, and managing any multi-factor authentication protocols.

By serving as the integration point for all system components, sensors, output devices, user interface, and network communication, the ESP32 ensures real-time operation, security, and adaptability within the biometric access control architecture.

2.5. User Interface and Feedback Components

The user interface (UI) and feedback components are essential for making biometric access control systems intuitive, accessible, and easy to use. These elements provide visual and auditory cues that guide users through the authentication process, confirm successful actions, or alert them to errors. A well-designed interface not only enhances the user experience but also improves system efficiency by reducing confusion and interaction time.

In this project, the user interface has been kept simple and functional by incorporating the following components: a liquid crystal display (LCD) for real-time messages and instructions, LED indicators (green and red) for visual feedback, a buzzer for auditory alerts, and a 4×3 keypad for user input such as PIN entry or navigation commands. Each of these components plays a specific role in facilitating smooth and effective interaction between the user and the biometric system. Their functionalities are detailed in the following subsections.

2.5.1. LCD Display

The LCD module (Figure 2.7) is a widely used user interface component in embedded systems due to its simplicity, low power consumption, and ease of integration. It contains an onboard controller and the necessary electronic components to manage text output, making it well-suited for real-time communication with users. LCDs are available in various configurations, typically ranging from 1 to 4 lines and 8 to 80 characters per line, with several models supporting different character formats and voltage levels.

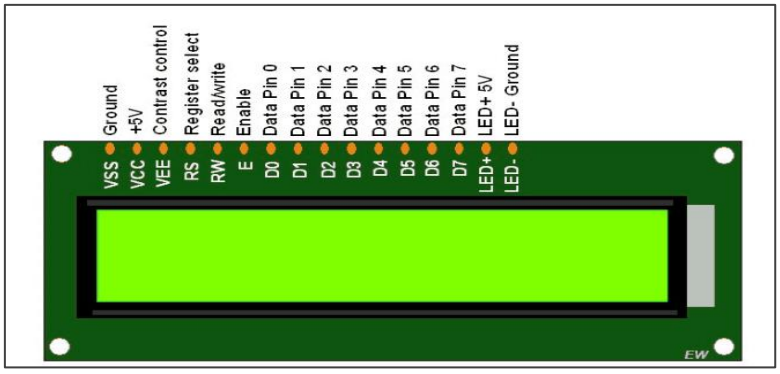


Figure 2.7: LCD 2×16 Display Module **[64].**

In this project, a 2x16 character LCD is used, which allows the display of two lines with up to 16 characters per line. This module is responsible for showing system messages such as "Place Finger," "Access Granted," or "Access Denied," providing real-time feedback to the user during interaction with the biometric system.

Many LCDs include a backlight feature, which enhances visibility in low-light conditions. The backlight is typically powered by LEDs, and the current required can vary from 80 mA to 250 mA, depending on brightness settings and the model. Proper current-limiting resistors or transistor drivers are recommended when integrating backlit LCDs to ensure safe operation and optimal display brightness **[63].**

2.5.2. The light-emitting diode LED

A light-emitting diode (LED) (Figure 2.8) is a semiconductor device that emits light when an electric current passes through it. LEDs are highly energy-efficient, long-lasting, and commonly used in embedded systems for status indication, signaling, and visual feedback.

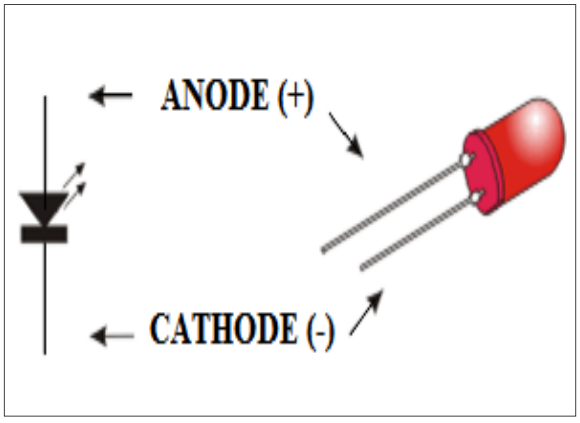


Figure 1.5.1.2.8: LED Pin Configuration **[65].**

In this biometric access control system, two LEDs are employed: a green LED to indicate successful authentication ("Access Granted"), and a red LED to signal a failed attempt or error ("Access Denied"). These visual cues provide immediate and intuitive feedback to the user.

The LEDs are controlled via the ESP32's GPIO (General Purpose Input/Output) pins. To safely interface an LED with the ESP32, the cathode (negative pin) of the LED is connected to GND, while the anode (positive pin) is connected in series with a 220-ohm resistor, which is then connected to an available GPIO pin. This resistor limits the current flowing through the LED, ensuring safe operation and protecting both the LED and the microcontroller **[66].**

This configuration enables the microcontroller to toggle the LEDs ON or OFF based on the access control logic, providing real-time feedback in response to fingerprint authentication outcomes.

2.5.3. Buzzer

A buzzer (Figure 2.9) is an electronic sound-emitting component that produces an audible tone when supplied with electrical power. Buzzers are commonly used in embedded systems to provide auditory feedback or alerts, such as notifications, warnings, or confirmation of actions.

Figure 2.9: Buzzer Module **[67].**

In the context of this biometric access control system, the buzzer is used to signal access outcomes: a short beep indicates a successful authentication ("Access Granted"), while a longer beep signals an error or denial ("Access Denied"). This form of feedback enhances usability by enabling users to recognize system status without needing to look at a display.

The buzzer operates with a typical supply voltage range of 3.3V to 5.0V, making it compatible with the ESP32's power outputs. For interfacing, the positive terminal of the buzzer is connected to a GPIO pin on the ESP32, while the negative terminal is connected to ground (GND). Through simple digital signals from the microcontroller, the buzzer can be activated or deactivated as needed **[67].**

This straightforward integration provides effective and immediate auditory cues, improving the overall user experience of the biometric system.

2.5.4. 4x3 Keypad

A 4×3 matrix keypad is a commonly used input device in embedded systems, particularly for applications involving user authentication, menu navigation, or function selection. When connected to a microcontroller such as the ESP32, it provides a simple and reliable interface for user interaction.

The keypad consists of 12 pushbuttons arranged in a 4-row by 3-column matrix, allowing it to detect and process individual keypresses efficiently. Typical use cases include entering user IDs, PIN codes, selecting system options, or triggering specific functions via key combinations.

The working principle of a matrix keypad involves scanning the rows and columns to determine which key has been pressed. In this configuration, the columns are usually configured as outputs, and the rows as inputs. Each button in the matrix connects a specific row to a column. When a key is pressed, it completes the circuit between a corresponding row and column, allowing the microcontroller to detect the key's position.

For example, if key '1' is pressed, the connection between its assigned row and column C1 will be completed. The ESP32 performs scanning logic to read the row input state after sequentially enabling each column output. This technique is illustrated in Figure 2.10, which shows the internal wiring of a typical 4×3 keypad **[68].**

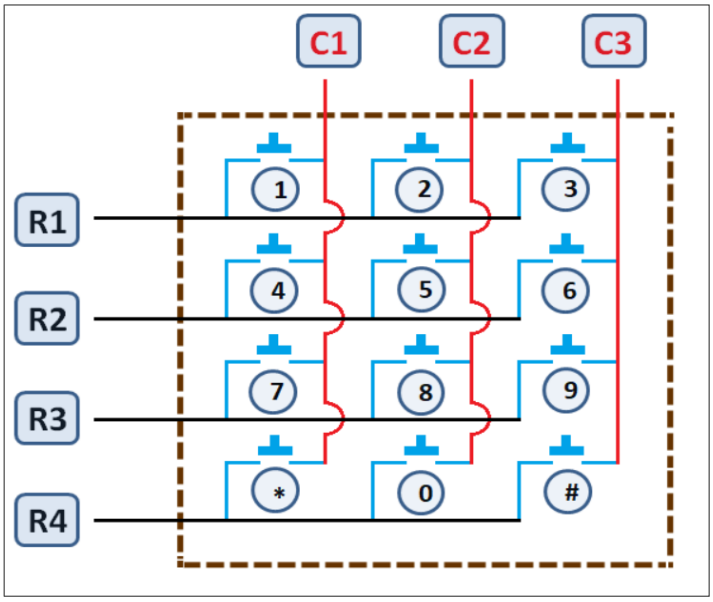


Figure 1.5.1.2.10: 4×3 Keypad Internal Circuit **[69].**

This method of input not only reduces the number of required GPIO pins but also provides a versatile interface for various user-driven functions within the biometric access control system.

2.6. Power Supply System

The biometric access control system is powered using a lithium iron phosphate ($LiFePO_4$) battery (Figure 2.11), which is well-suited for embedded applications due to its stability, efficiency, and safety characteristics. This type of battery has a nominal voltage of 3.2V and a maximum fully charged voltage of 3.65V.

Figure 1.5.1.2.11: Lithium Iron Phosphate ($LiFePO_4$) Battery **[71].**

One of the key advantages of the $LiFePO_4$ battery is its very flat discharge curve, meaning that its output voltage remains nearly constant over most of the discharge cycle. This ensures stable power delivery to the components, minimizing voltage fluctuations that could affect performance.

The maximum voltage of 3.65V is only slightly higher than the ESP32's maximum rated operating voltage of 3.6V, making the $LiFePO_4$ battery a compatible and efficient power source. In many cases, it can be connected directly to the 3.3V power input pin of the ESP32 without the need for complex voltage regulation circuitry **[70].**

This choice of power supply enhances the system's portability and makes it suitable for standalone or off-grid installations, where reliable and long-lasting power is essential.

2.7. Communication Interfaces

To ensure reliable data exchange between components in the biometric access control system, the design incorporates two communication protocols.

This section describes the protocols used: UART (Universal Asynchronous Receiver-Transmitter) and Wi-Fi (Wireless Fidelity).

2.7.1. UART (Universal Asynchronous Receiver-Transmitter)

UART is a widely used asynchronous serial communication protocol that facilitates data exchange between two digital devices. Unlike synchronous protocols such as SPI or I^2C, UART does not require a clock line to synchronize communication. Instead, both devices must agree on a common baud rate, which defines the speed of data transmission in bits per second.

In UART communication (Figure 2.12), data is transmitted bit by bit over two lines: one for transmitting (TX) and one for receiving (RX). This simplicity makes UART highly suitable for point-to-point communication in embedded systems **[72].**

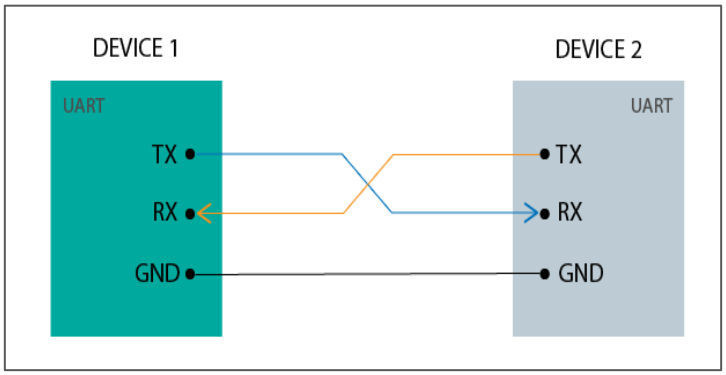


Figure 1.5.1.2.12: UART Communication Protocol **[72].**

The ESP32 microcontroller supports up to three UART interfaces: UART0, UART1, and UART2, depending on the specific board configuration. The GPIO assignments for each UART port are as follows:

- **UART0**: GPIO1 (TX0), GPIO3 (RX0) Commonly used for communication with the serial monitor during programming and debugging. It can be reassigned to other peripherals if not required for console output.
- **UART1**: GPIO10 (TX1), GPIO9 (RX1), typically used for external device communication, such as with sensors or secondary microcontrollers.
- **UART2**: GPIO17 (TX2), GPIO16 (RX2), also commonly used for connecting to external peripherals, such as the fingerprint sensor in this project **[72].**

2.7.2. Wi-Fi (Wireless Fidelity)

Wi-Fi is a wireless communication technology that allows devices to connect to a local area network (LAN) or the Internet without physical cabling. It uses radio frequency (RF) signals to transmit data between devices and a central access point, typically a router. This technology is widely adopted in embedded systems for applications requiring remote monitoring, cloud connectivity, or web-based control interfaces.

The ESP32 microcontroller supports the IEEE 802.11 b/g/n Wi-Fi standard, operating in the 2.4 GHz frequency band. This built-in capability enables the ESP32 to connect to wireless networks (Figure 2.13), communicate with servers, and transmit biometric or access data securely and in real time **[73].**

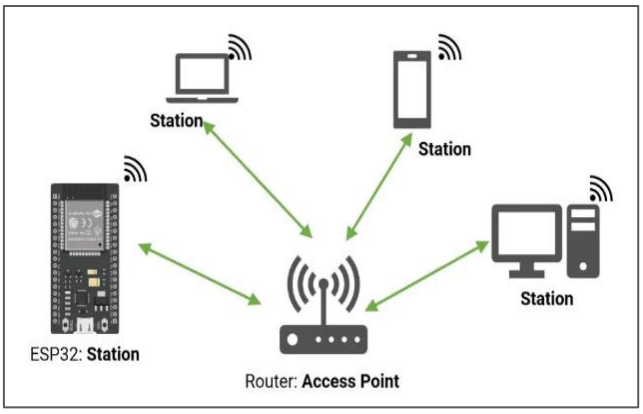


Figure 1.5.1.2.13: Connecting ESP32 to Wi-Fi **[74].**

The ESP32 supports three primary Wi-Fi operating modes, providing flexibility for different network topologies and use cases **[74]:**

- **Mode 1: Wi-Fi Station Mode (STA)**: In this mode, the ESP32 connects to an existing wireless network, such as a Wi-Fi router. Upon successful connection, the router assigns a unique IP address to the ESP32, enabling it to send or receive data over the local network or internet.
- **Mode 2: Access Point (AP) Mode**: The ESP32 can also function as a wireless access point, creating its own Wi-Fi network. In this mode, devices such as smartphones, tablets, or laptops can connect directly to the ESP32, allowing local communication without a router.
- **Mode 3: Dual Mode (STA + AP)**: In this mode, the ESP32 operates as both a station and an access point simultaneously. This allows it to connect to a router while also allowing other devices to connect directly to it, enabling both internet access and local peer-to-peer communication.

These Wi-Fi capabilities make the ESP32 particularly valuable for cloud-based access control, real-time logging, firmware updates, and mobile device interaction in biometric systems.

2.8. Actuation Mechanism

The actuation mechanism in a biometric access control system refers to the physical device responsible for executing a mechanical action, such as unlocking a door or releasing an electric lock following successful biometric authentication (e.g., fingerprint, facial recognition, or iris scan). This mechanism forms the final step in the access control process, translating digital verification results into real-world physical responses.

In this project, the actuator used to grant or deny access is a servo motor, chosen for its precise control and ease of integration with microcontrollers such as the ESP32.

2.8.1. Servo motor

A servo motor (Figure 2.14) is a specialized electromechanical device designed for precise control of angular or linear motion. It operates using a closed-loop feedback mechanism, where the motor receives a control signal and adjusts its position accordingly. This allows the servo to rotate or move to an exact angle or position, making it ideal for applications that demand accuracy, reliability, and smooth movement, such as in automated locks and access gates **[75].**

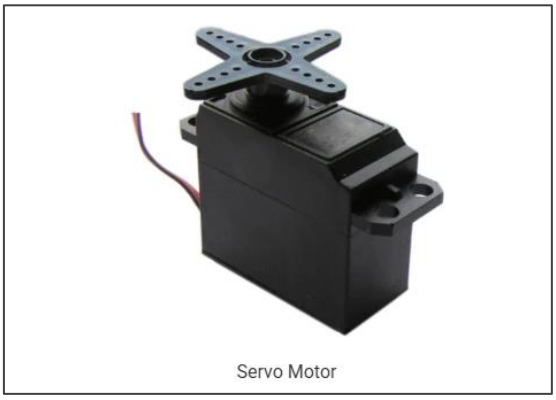


Figure 2.14: Servo motor **[75].**

In the context of this access control system, the servo motor is used to physically open or close the locking mechanism upon receiving an authenticated signal from the ESP32.

The integration of the servo motor with the ESP32 microcontroller is straightforward, involving three primary connections:

- **+VCC (Red Wire)**: Connects to a 5V power supply. For the commonly used SG90 micro servo, the operating voltage is typically 4.8V to 5V.
- **Ground (Brown Wire)**: Connects to the ground (GND) pin of the ESP32 to complete the electrical circuit.
- **Control Signal (Orange Wire)**: Connects to a PWM-capable GPIO pin on the ESP32. The control signal typically consists of a pulse-width modulation (PWM) signal with a 20ms period (50 Hz). The pulse width determines the target angle or position of the servo.

The ESP32 can generate this PWM signal either through software libraries (e.g., Arduino Servo library) or hardware PWM modules. Upon receiving the control signal, the servo adjusts its shaft to a specific angle, unlocking or locking the door as instructed **[75].**

2.9. Conclusion

This chapter has presented a comprehensive overview of the hardware design for the fingerprint-based biometric access control system, emphasizing the functionality and interdependence of each component in achieving secure and efficient operation. The selection of the ESP32 microcontroller was guided by its optimal balance of processing performance, connectivity options, and flexibility, making it particularly well-suited for real-time biometric authentication tasks.

The integration of the fingerprint sensor module facilitates accurate and user-friendly biometric identification, while auxiliary components, such as the power supply, user interface, communication interfaces, and actuation mechanism, enhance the system's overall reliability and usability. Each hardware element was chosen based on its compatibility, efficiency, and contribution to the system's objectives.

Through thoughtful selection and seamless integration of these components, the system is capable of capturing biometric data, processing it efficiently, and controlling access mechanisms with a high degree of accuracy and responsiveness. This hardware infrastructure lays the foundation for a secure and scalable biometric solution.

The next chapter will transition to the software implementation, where system logic, data processing algorithms, and embedded security features will be examined in detail to complement the hardware framework established in this section.

Chapter3: Software design of the biometric access control system

Chapter 3: Software Design of the Biometric Access Control System

3.1. Introduction

While hardware provides the structural foundation of any biometric access control system, it is the software layer that drives system intelligence, operational logic, and interaction with users and external services. This chapter presents the software design and implementation aspects of the fingerprint-based biometric access control system introduced earlier.

The software is responsible for handling biometric data acquisition, user authentication logic, interface control, and communication with peripheral devices and remote services. In this context, the ESP32 microcontroller runs firmware developed using the Arduino IDE, chosen for its simplicity, extensive library support, and compatibility with the selected hardware modules.

This chapter begins by outlining the overall software architecture and the development environment, followed by a detailed explanation of the programming languages used, particularly within the Arduino ecosystem. The chapter then describes the structure of an Arduino sketch, including library inclusion, variable declaration, and the core setup() and loop() functions. Special emphasis is placed on the integration of key software libraries that enable fingerprint recognition, wireless communication, cloud database interaction, HTTP protocols, and real-time functions.

In addition to code development, this chapter also discusses simulation and testing tools such as Proteus ISIS, which are used to validate design logic and simulate hardware behavior. The chapter concludes by examining the integration of Firebase, a cloud-based real-time database platform, and the use of HTTP protocols to enable secure communication between the biometric system and remote servers.

By the end of this chapter, the reader will have a comprehensive understanding of the software elements that transform the system's hardware into a functional, secure, and responsive biometric access control solution.

3.2. Software Architecture Overview

The proposed software architecture for the biometric access control system is illustrated in Figure 3.1. It is designed to enhance physical security by utilizing fingerprint recognition technology for user verification and access authorization. The software design integrates several

coordinated modules, each responsible for a specific function in the overall authentication process.

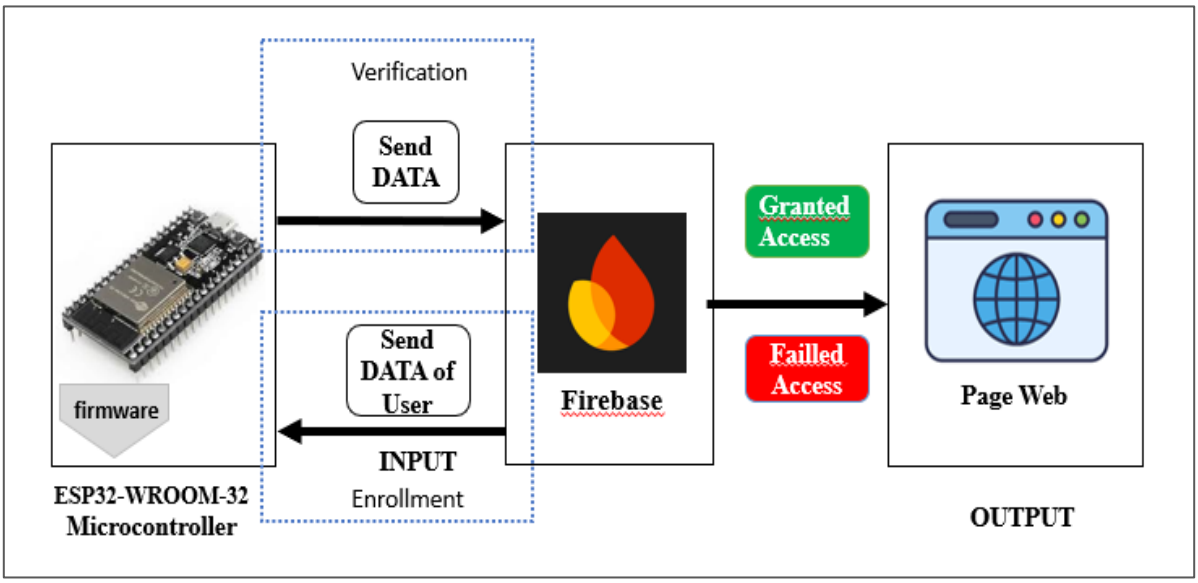


Figure 3.1: Software architecture of the biometric access control system.

At the core of the system is the ESP32-WROOM-32 microcontroller, which executes embedded firmware responsible for managing the biometric sensor, collecting fingerprint data, processing enrollment and verification logic, and handling communication with external services. The ESP32 also manages user interface elements and triggers output actions such as access status updates.

The software system operates in two key phases: enrollment and verification. During enrollment, the user's fingerprint and identity data are registered in Firebase, a cloud-based real-time database. This data is later used for comparison during the verification phase. When a user attempts to gain access, the ESP32 captures the fingerprint data, transmits it to Firebase, and awaits a response. Based on the verification result returned from Firebase, access is either granted or denied.

The final output of the system is displayed through a web interface, which can show real-time status such as "Access Granted" or "Access Denied." The system can also be extended to include additional output mechanisms such as Telegram notifications or mobile alerts, leveraging the ESP32's built-in Wi-Fi and Bluetooth connectivity for future enhancements and integrations with cloud platforms.

The system's structure can be divided into three main logical components:

- **Input**: Firebase (user registration and storage)

- **Processing Unit**: ESP32-WROOM-32 (biometric data handling, communication, and decision-making)
- **Output**: Firebase (as a middleware), Web page, and optional notification channels

This modular software architecture ensures flexibility, scalability, and responsiveness in real-time biometric access applications.

3.3. Development Tools and Environment

The successful implementation of the biometric access control system relies heavily on a well-structured software development environment and the appropriate selection of tools. This section presents the key technologies, platforms, libraries, and simulation environments used to design, develop, and test the embedded software for the ESP32 microcontroller.

The system's firmware was developed using the Arduino IDE, which supports C++ as its primary programming language. Several open-source libraries were integrated to handle key tasks such as fingerprint recognition, cloud communication, user interface control, and real-time clock functionality. Additional tools such as Proteus were used for hardware simulation and verification, while Firebase Console served as the cloud-based backend for user data and authentication logs. The system also includes a web interface, developed using HTML and CSS, to display access status and provide a remote monitoring option.

3.3.1. Programming language (C++)

The firmware for the biometric access control system was developed using C++, a high-level, object-oriented programming language widely used in embedded systems. C++ provides the necessary low-level control over hardware resources while supporting modular and structured code development. This makes it ideal for writing firmware that interacts directly with microcontroller peripherals and external hardware modules.

Within the Arduino IDE environment, C++ is used to define all core functionalities of the system, including sensor interfacing, input/output control, serial communication, and decision-making logic. The object-oriented nature of C++ enables efficient management of hardware modules by encapsulating related operations within classes or structured functions. For example, operations involving the fingerprint sensor, LCD screen, and keypad are abstracted into manageable code blocks using available libraries written in or compatible with C++.

C++ also facilitates efficient memory usage, interrupt handling, and real-time task management—critical aspects in a biometric access system where response time, reliability,

and data accuracy are essential. Its compatibility with a wide range of microcontroller architectures, including the ESP32, further underscores its suitability for this project **[76]**.

3.3.2. Development platform (Arduino IDE) for firmware

The Arduino Integrated Development Environment (IDE) (Figure 3.2) was selected as the primary platform for developing the firmware of the biometric access control system. The Arduino IDE provides a user-friendly interface and a robust set of tools for writing, compiling, uploading, and debugging C++ code on microcontroller-based platforms, including the ESP32-WROOM-32 used in this project.

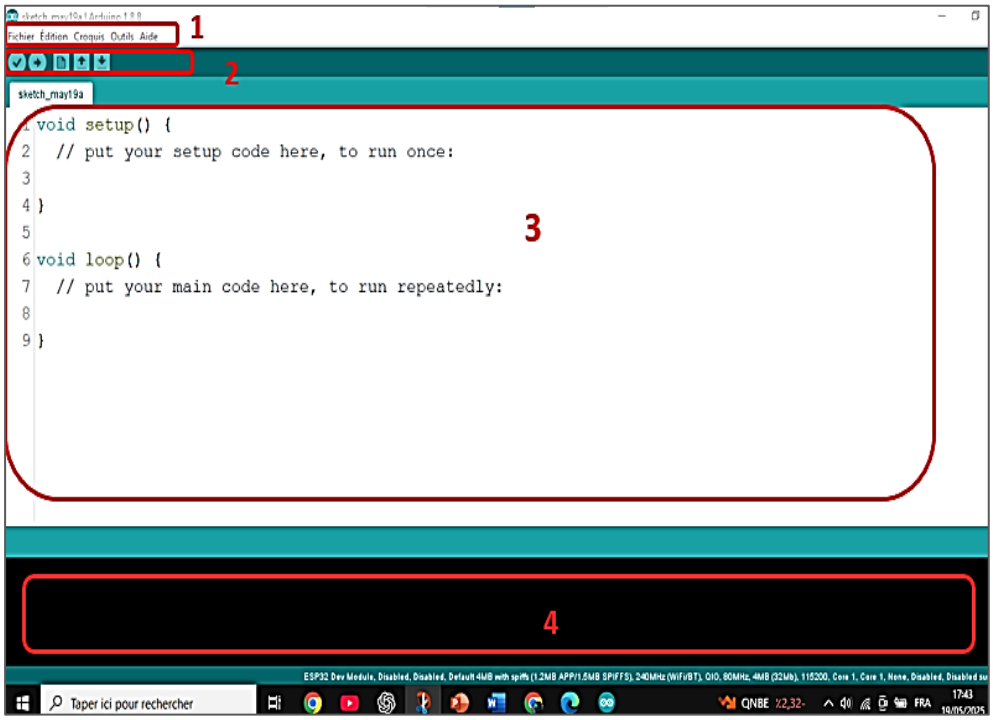


Figure 3.2: Interface of Arduino IDE.

One of the major advantages of the Arduino IDE is its simplicity and accessibility, making it suitable for both rapid prototyping and full-scale embedded development. It supports a wide range of microcontrollers and provides extensive compatibility with numerous open-source hardware libraries, which significantly streamlines the integration of external modules such as fingerprint sensors, LCD displays, keypads, and Wi-Fi connectivity.

The IDE includes essential development tools such as:

- A code editor with syntax highlighting and error checking.
- An integrated compiler and linker to convert C++ source code into executable machine code.

- A serial monitor for real-time communication and debugging with the microcontroller via UART.

In the context of this system, the Arduino IDE was used to write and manage all firmware functions. The development process included initializing hardware components, managing input/output operations, processing biometric authentication logic, and establishing communication with Firebase and the web interface. Furthermore, the availability of community-contributed libraries for the ESP32 simplifies access to advanced features like Wi-Fi configuration, HTTP requests, and real-time database interactions.

Additionally, the Arduino IDE supports cross-platform compatibility (Windows, macOS, Linux), and its lightweight footprint makes it easy to deploy on a wide range of development machines. The combination of flexibility, community support, and integrated tools makes the Arduino IDE a practical and efficient choice for the development of embedded access control systems **[77].**

3.3.2.1. Arduino IDE Editing Window

The Arduino IDE editing window is the central workspace where developers write and manage their source code. It is visually organized into four main zones, each designed to support a specific aspect of the programming workflow **(Figure 3.2).**

- **Zone 1: Menu Bar**: This section provides access to software configuration options such as board selection, library management, and port configuration.
- **Zone 2: Toolbar**: This area includes shortcut buttons for essential actions such as verify (compile), upload (send code to the board), open, save, and launch the serial monitor.
- **Zone 3: Code Editor**: This is the main text area where users write the source code. It supports C++ syntax highlighting and basic error indication.
- **Zone 4: Output Console (Debugger)**: This section displays compiler messages, error diagnostics, and serial communication outputs, making it a valuable tool for troubleshooting and debugging.

3.3.2.2. Structure of an Arduino IDE Program

An Arduino program, also referred to as a sketch, is typically composed of three key components:

a) Declaration of Libraries and Variables

At the beginning of the sketch, relevant libraries and global variables are declared. Libraries are pre-written sets of functions written in C/C++ that simplify complex tasks, such as interacting with sensors or managing communication protocols.

Variables, on the other hand, serve as containers to store data values during program execution. Common variable types include int, float, char, String, double, and bool. Proper variable declaration is essential for memory management and logic control.

b) setup() Function

The setup() function is executed once, when the microcontroller is powered on or reset. It is used to initialize configurations such as pin modes, communication interfaces, or library objects.

c) loop() Function

Following the setup, the loop() function is executed continuously, allowing the program to react to inputs, update outputs, and perform ongoing tasks. It forms the core logic of the Arduino sketch and enables dynamic control of the hardware.

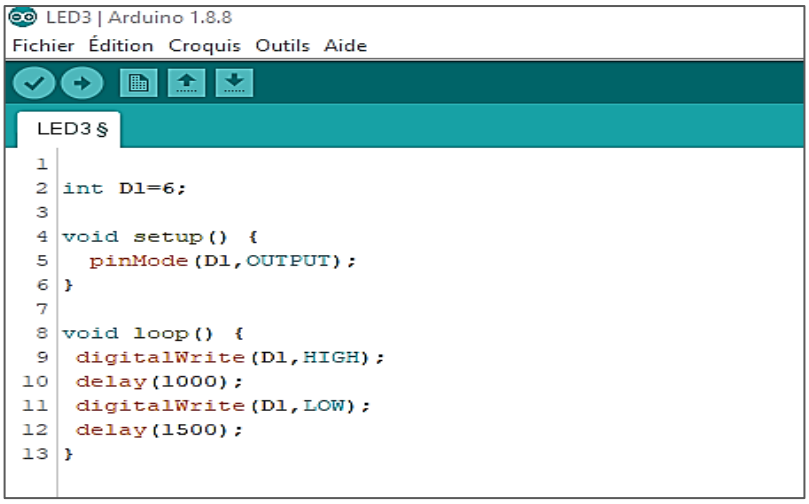


Figure 3.3: Example Arduino Code: Blink an LED.

The figure 3.3 provide an illustration of simple program makes an LED connected to pin 6 blink repeatedly.

3.3.3. Key Libraries used in Biometric Access Control System

In embedded systems development, libraries provide a powerful abstraction layer that simplifies the interaction with hardware and external services. A library is a collection of pre-

written code, typically composed of functions, classes, and constants, that developers can use to perform common or complex tasks without having to write the code from scratch.

In this biometric access control system, several key libraries were employed to manage specific modules, including the fingerprint sensor, Wi-Fi connectivity, Firebase integration, HTTP communication, time management, and user interface elements.

3.3.3.1. Fingerprint Library

The Adafruit Fingerprint Sensor Library (Figure 3.5) is an open-source library designed specifically to interface with Adafruit optical fingerprint sensors. It supports functions such as capturing fingerprint images, enrolling new users, verifying identity, and deleting stored fingerprints. The library communicates with the sensor via TTL serial interface and can store up to 162 fingerprint templates in the sensor's onboard flash memory.

```
4 #include <Adafruit_Fingerprint.h>
```

Figure 3.4: Adafruit Fingerprint Library in Arduino IDE.

By using the Adafruit_Fingerprint.h library, developers can quickly integrate biometric authentication features into Arduino-based projects, significantly reducing development time and complexity. It includes example sketches for enrollment, search, and template management **[80].**

3.3.3.2. Wi-Fi Library

The WiFi library (Figure 3.5) enables the ESP32 to connect to wireless networks and the Internet. It provides comprehensive functionality for scanning nearby networks, connecting to a selected SSID, managing connection status, and obtaining IP configurations.

```
10 #include <WiFi.h>
11 // ⬘ Informations WiFi
12 const char* ssid = "    ";
13 const char* password = "    ";
```

Figure 3.5: Wi-Fi Library in Arduino IDE.

This library is crucial in allowing the ESP32 to function as a client device that interacts with cloud services and web interfaces. The developer can specify the SSID and password to establish a secure Wi-Fi connection **[81].**

3.3.3.3. Firebase Library

The Firebase_ESP_Client library (Figure 3.6) is a high-performance library developed for interfacing the ESP32 microcontroller with Firebase Realtime Database. It supports a wide range of operations including reading, writing, updating, and deleting data.

```
2 #include <Firebase_ESP_Client.h>
```

Figure 3.6: Firebase Library in Arduino IDE.

Additionally, the library allows real-time event listening, authentication with Firebase credentials, and security policy management. It is optimized for low-latency and secure cloud communication, making it ideal for access control applications where real-time database synchronization is needed **[82].**

3.3.3.4. HTTP Client Library

The HTTPClient library (Figure 3.7) is part of the official Arduino core for ESP32 and allows the microcontroller to perform HTTP and HTTPS requests. It is used to send or receive data from web servers or cloud APIs.

```
6 #include <HTTPClient.h>
```

Figure 3.7: HTTP Client Library in Arduino IDE.

This library is especially useful when implementing RESTful API communication or interacting with external systems through standard web protocols **[83].**

3.3.3.5. Time Library

The time.h library (Figure 3.8) is used to manage time-based functions in the ESP32. When paired with NTP (Network Time Protocol) services, it can synchronize the internal clock with Internet-based time servers.

```
8 #include <time.h>
9 #include <NTPClient.h>
```

Figure 3.8: Time Library in Arduino IDE.

This functionality is important for timestamping access events, logging activity, or scheduling periodic operations in real time **[84].**

3.3.4. Proteus for circuit simulation and testing

Proteus (Figure 3.9) is a widely used electronic design automation (EDA) software suite developed by Labcenter Electronics. It is primarily used for schematic capture, circuit simulation, and PCB (printed circuit board) design. Proteus is well-known in both academic and industrial environments and is extensively used in engineering education for virtual prototyping and system testing.

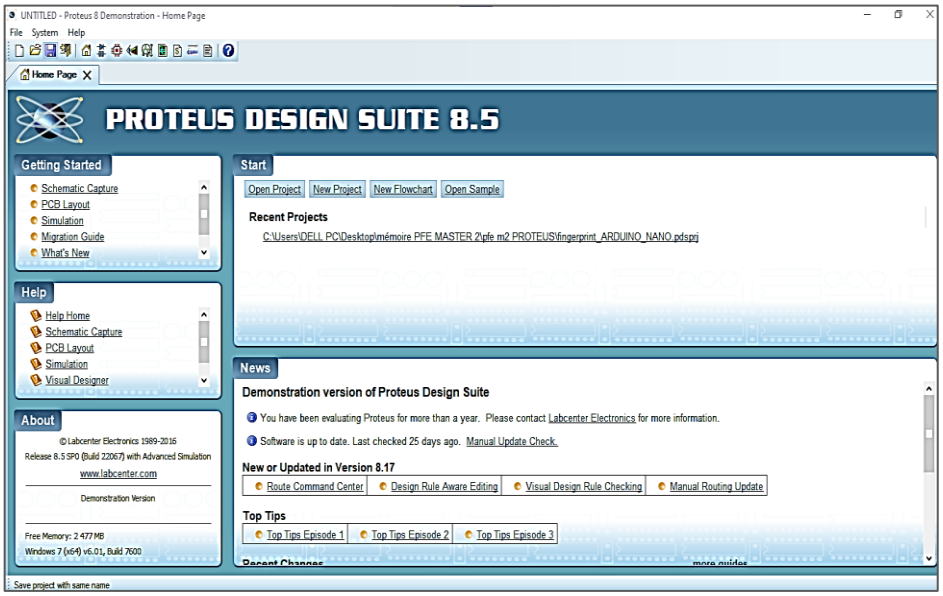


Figure 3.9: Proteus Software Suite Interface.

The Proteus suite comprises two core modules :

- ISIS (Intelligent Schematic Input System) for schematic design and simulation
- ARES (Advanced Routing and Editing Software) for PCB layout and routing

By using Proteus, developers can simulate embedded systems, microcontroller programs, and complete circuit behavior before physically implementing the design. This approach saves time, reduces hardware costs, and allows early detection of logical and electrical errors **[85].**

3.3.4.1. ISIS: Schematic Design and Simulation

The ISIS module of Proteus (Figure 3.11) is primarily used for schematic design and simulation of electrical and electronic circuits. It provides a graphical interface where components can be added, connected, and simulated in real time. This capability is particularly useful in verifying the logical behavior of embedded systems, including the biometric access control system developed in this project.

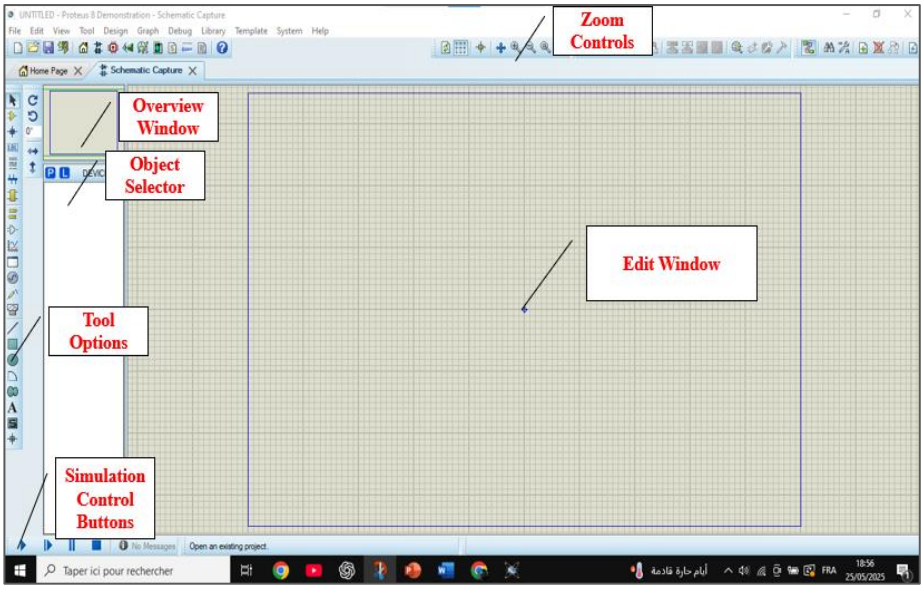


Figure 3.10: Main Window of Proteus ISIS.

The main interface of Proteus ISIS is divided into several functional areas:

- **Edit Window**: The central, dotted region where circuits are drawn and simulations are executed. It serves as the primary workspace for project design.
- **Overview Window**: Provides a zoomed-out view of the entire schematic, allowing users to navigate large designs efficiently.
- Object Selector (P and L Buttons):
    - P Button: Opens the component selection window, where electronic parts (resistors, microcontrollers, sensors, etc.) can be added to the design.
    - L Button: Allows modification of component properties, such as resistance values or pin labels.
- **Zoom Controls**: Enable users to zoom in and out of the schematic to inspect and fine-tune circuit layout and wiring.
- **Tool Options**: This toolbar includes simulation instruments such as voltmeters, ammeters, and oscilloscopes, which can be added to circuits for real-time measurement and analysis.
- **Simulation Control Buttons**: Located in the lower-left corner, these include Run, Pause, Stop, and Restart. These buttons control the simulation flow, allowing users to test and debug the circuit under different conditions.

By simulating the system's hardware connections and logic in Proteus ISIS before deployment, design flaws and integration issues can be identified early, reducing development time and improving system reliability.

3.3.5. Firebase Console for cloud integration

Firebase (Figure 3.11) is a comprehensive platform developed by Google that offers a suite of cloud-based services tailored to support the development of high-quality web and mobile applications. It enables developers to focus on front-end implementation without the need to manage complex backend infrastructure.

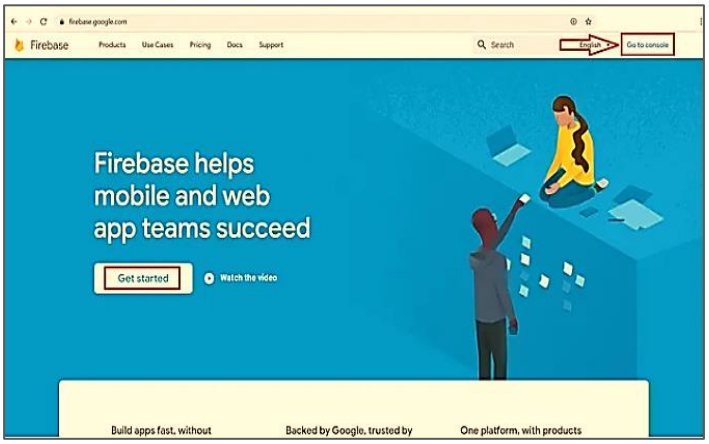


Figure 3.11: Firebase database interface **[86].**

Firebase provides a wide range of functionalities, including:

- Realtime Database
- User authentication
- Cloud storage
- Analytics
- In-app messaging
- Machine learning services

The platform supports cross-platform development, including Android, iOS, and web-based applications. It streamlines development by enabling serverless architecture, allowing developers to build responsive applications with minimal setup and maintenance **[86].**

3.3.5.1. Firebase vs. Traditional Databases

Firebase represents a shift from traditional relational databases by offering a NoSQL-based real-time backend that is optimized for scalable and reactive applications. Traditional databases

such as MySQL, PostgreSQL, and Oracle use structured schemas with tables, columns, and rows, which are ideal for applications requiring complex relational queries and strict transaction management.

In contrast, Firebase stores data in a JSON tree structure, enabling real-time data synchronization across multiple clients. It is particularly well-suited for applications that require immediate updates across connected devices, such as chat systems, collaborative tools, and biometric access logs.

A comparative summary is shown in Figure 3.12:

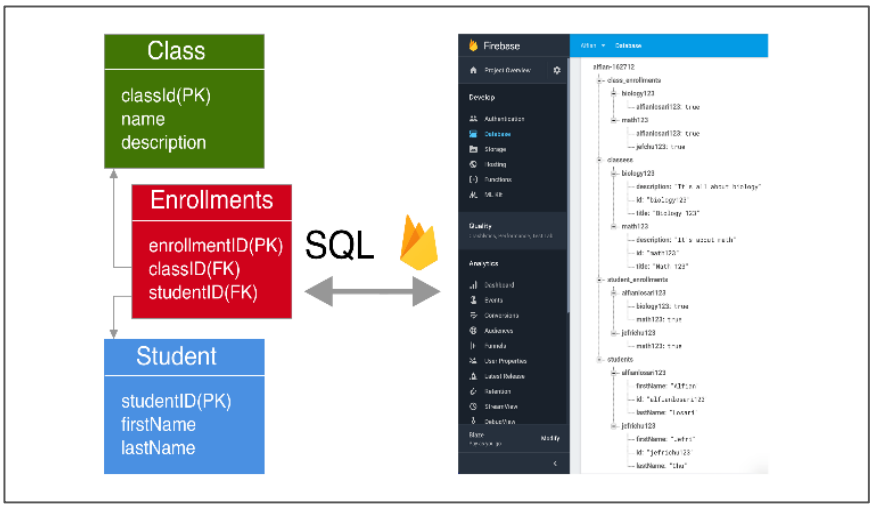


Figure 3.12: Comparison Between Firebase and Traditional Databases **[86].**

3.3.5.2. Firebase Realtime Database

The Firebase Realtime Database is a cloud-hosted NoSQL database that stores data in JSON format. It synchronizes data across all connected clients in real time, making it ideal for systems that require instant access to shared data, such as biometric authentication logs and access control status.

```
void enregistrerEvenementFirebase(String statut, int userId) {
    String path = "/evenements/" + String(millis());

    FirebaseJson json;
    json.set("statut", statut);
    json.set("userId", userId);
    json.set("timestamp", millis());
    json.set("date_heure", getDateHeure());

    if (Firebase.RTDB.pushJSON(&fbdo, path.c_str(), &json)) {
        Serial.println("📩 Données envoyées à Firebase !");
    } else {
        Serial.println("❌ Erreur Firebase !");
        Serial.println(fbdo.errorReason());
    }
}
```

Figure 3.13: Program Using Firebase Realtime Database **[87].**

When using Firebase's cross-platform SDKs (for Android, iOS, and JavaScript), all clients share a single instance of the Realtime Database. Any changes made to the data are instantly pushed to all other clients, enabling seamless updates and data consistency across devices.

In this project, the Firebase Realtime Database is used to:

- Register user credentials
- Store fingerprint authentication results
- Log access attempts in real time
- Synchronize data between ESP32 and web dashboard

A sample implementation using the Firebase Realtime Database is shown in Figure 3.13:

3.3.6. HTML/CSS for web interface

To enable remote access and real-time monitoring of the biometric access control system, a web interface was developed using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). These technologies form the structural and visual foundation of modern web applications.

HTML is used to define the structure and content of web pages. In this project, HTML elements are used to organize and display system messages such as "Access Granted," "Access Denied," and user activity logs. Elements like headings, paragraphs, tables, and divisions allow the interface to be structured clearly for user interaction.

CSS complements HTML by providing styling and layout control. It is used to enhance the visual presentation of the interface, ensuring a responsive and user-friendly design that can be accessed from any modern browser. Features such as color coding, font adjustments, and layout grids help convey access status in an intuitive manner.

The web interface is connected to the Firebase Realtime Database, allowing it to automatically update its content as new authentication results are pushed from the ESP32 microcontroller. This enables real-time feedback and system transparency for administrators and users alike.

In addition, the interface is accessed over the Internet using HTTP (Hypertext Transfer Protocol), a standard web protocol that governs communication between web browsers (clients) and servers. HTTP enables the retrieval and transmission of data such as access logs, user

information, and control signals. It serves as the underlying transport mechanism for web-based interaction (Figure 3.14).

Together, HTML, CSS, and HTTP provide a robust platform for delivering biometric system feedback and control functions through a clean, interactive, and browser-accessible interface.

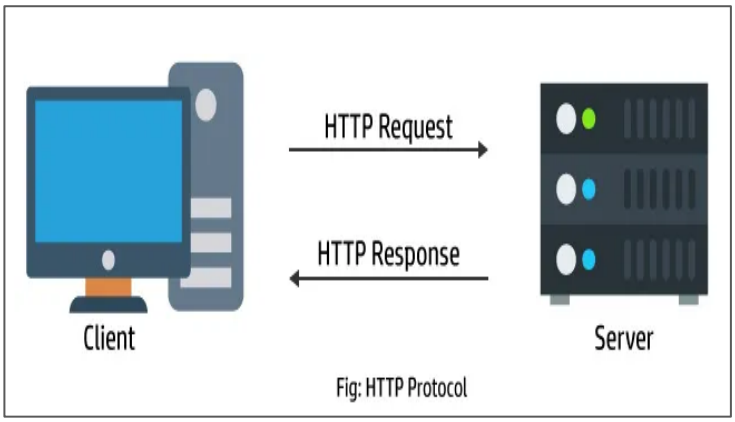


Figure 3.14: HTTP (Hypertext Transfer Protocol) Conceptual Diagram **[88]**.

3.4. Conclusion

This chapter has presented the complete software design and implementation of the fingerprint-based biometric access control system. The development process integrated a robust set of tools and technologies, including the Arduino IDE, C++ programming language, and multiple specialized libraries to support fingerprint recognition, cloud communication, real-time monitoring, and user interface management.

The ESP32 microcontroller served as the central processing unit, executing firmware developed using modular and reusable code structures. External libraries such as Adafruit_Fingerprint, Firebase_ESP_Client, and HTTPClient greatly simplified the integration of complex functionalities. Additionally, Proteus ISIS was used to simulate and verify circuit behavior, enhancing system reliability during the development phase.

The system's connection to Firebase Realtime Database provided a secure and scalable backend for real-time user authentication and data logging. The implementation of a simple yet effective web interface using HTML and CSS allowed administrators to monitor system activity remotely, reinforcing usability and accessibility.

Through the strategic selection of software tools and careful design of system logic, the software component effectively complements the hardware architecture outlined in Chapter 2. Together, they enable a complete, functional, and user-friendly biometric access control solution.

The following chapter will focus on the testing, evaluation, and performance analysis of the system, validating its effectiveness and identifying areas for potential improvement.

Chapter 4: Implementation Results and System Validation

Chapter 4: Implementation Results and System Validation

4.1. Introduction

This chapter presents the practical implementation and validation of the fingerprint-based biometric access control system developed in the previous chapters. After detailing the system's hardware design and software architecture, it is essential to evaluate how the system performs under real-world conditions and to assess its reliability, responsiveness, and effectiveness in securing access.

The implementation results are based on actual deployment and testing of the system using the ESP32 microcontroller, DY50 fingerprint sensor, LCD interface, keypad, and integration with cloud services such as Firebase and Telegram. The system was tested for a range of functional scenarios, including user enrollment, fingerprint verification, password validation, access authorization, and denial conditions.

Each software module, including Wi-Fi connectivity, Firebase real-time database operations, Telegram notifications, and real-time clock synchronization, was validated individually and in combination. Additionally, the web interface was tested to ensure that access events are correctly displayed and updated in real time. Proteus simulation was also used to verify the logical behavior of the hardware before final deployment.

This chapter also includes a discussion of observed performance metrics such as system latency and fingerprint recognition accuracy. Finally, the economic feasibility of the system is analyzed, and limitations encountered during implementation are addressed.

4.2. Experimental Setup

To evaluate the functionality and performance of the biometric access control system, a controlled experimental setup was established. The objective was to simulate a real world usage condition where users interact with the system for secure access, and to validate that all hardware and software components function cohesively and reliably.

In the physical prototype, all components were mounted on a prototyping breadboard. The ESP32-WROOM-32 microcontroller served as the central processing unit, orchestrating sensor input, data processing, and communication with cloud services. The DY50 optical fingerprint sensor was connected to the ESP32 via UART communication on GPIO16 (Rx) and GPIO17 (Tx). A 16×2 character LCD was used to display system prompts, statuses, and feedback, interfaced with six GPIO pins (13, 12, 15, 2, 18, and 4).

A 4×4 matrix keypad was used for user input, specifically for selecting actions and entering passwords. Its rows were connected to GPIOs 23, 32, 35, 34, and the columns to GPIOs 26, 25, 27, and 14. Visual access feedback was provided by green and red LEDs, connected to GPIO22 and GPIO23, respectively. A passive buzzer was optionally included for audible confirmation of access results. Power to the ESP32 was supplied either through a USB connection or a 3.7V $LiFePO_4$ battery, regulated internally to match ESP32 operating levels.

Figure 4.1 shows the hardware wiring diagram, created using Fritzing software, illustrating the interconnection between components and the power distribution strategy.

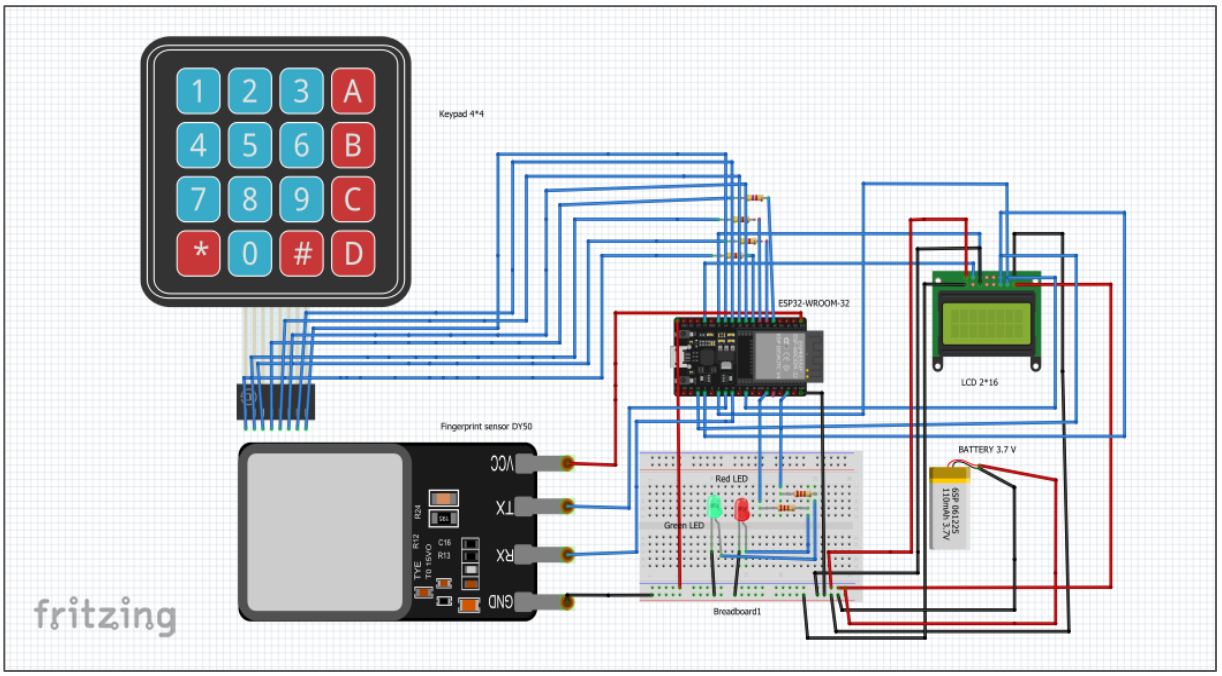


Figure 4.1: Schematic of physical hardware connection using ESP32 (generated with Fritzing software).

To simulate and test the circuit before real deployment, ISIS Proteus software was used. Since the ESP32 is not natively available in Proteus, an Arduino Nano was used as a placeholder to emulate the logical behavior of GPIOs and UART communication. All peripheral devices, including the keypad, LEDs, LCD, and fingerprint module, were integrated into the simulation circuit. This allowed verification of pin level behavior, input/output logic, and user interface responses without risking damage to physical components. Figure 4.2 displays the Proteus-based schematic, which helped validate the theoretical design and test initial behaviors such as serial communication, keypad input, and LCD messages.

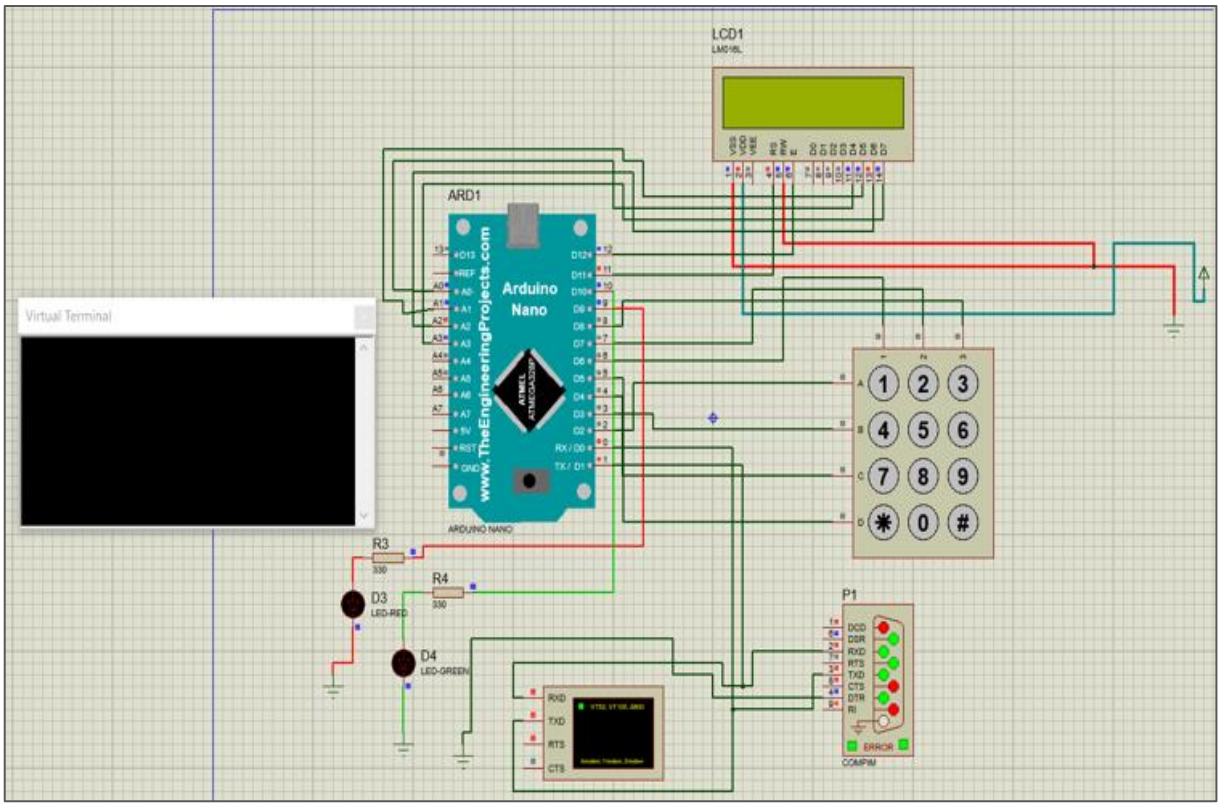


Figure 4.2: Schematic of physical hardware connection using ESP32 (generated with ISIS Proteus software).

Upon powering the physical system, the LCD shows a startup message ("System Started"), indicating system initialization. The ESP32 then attempts to connect to the predefined Wi-Fi network. Once connected, it displays ("WiFi Connected") and proceeds to initialize the fingerprint module. The NTP client synchronizes with time servers to ensure that all events are accurately timestamped. Following successful initialization, the main menu appears, offering users the option to enroll a fingerprint or verify their identity, allowing the system to operate interactively and in real time.

4.2.1. Software Operation and Behavior

The software developed for the biometric access control system was deployed to the ESP32-WROOM-32 microcontroller using the Arduino IDE. Written in C++, the firmware coordinates all system functions, including hardware initialization, user interface control, biometric data processing, and cloud communication via Firebase and Telegram.

Upon startup, the software initializes all hardware modules, starting with the LCD, fingerprint sensor, and keypad, before establishing a Wi-Fi connection using the stored SSID and password credentials. Once the device connects to the network, it configures access to the Firebase Realtime Database and starts the NTP client to synchronize the internal clock with

global time servers. This ensures that all access events are timestamped with precise date and time information.

After initialization, the system enters its main execution loop, where it remains responsive to user inputs via the 4×4 keypad. Two primary options are available:

- **[4] Enroll**: Initiates the fingerprint registration process (Enrollment mode).
- **[5] Verify**: Starts the fingerprint-based identity verification and access check (Verification mode).

In enrollment mode, the system prompts the administrator to enter the user's ID, name, contact, and password via the serial interface. It then captures two images of the user's fingerprint to create a reliable template. If successful, this template is stored locally on the sensor, and the user's data is uploaded to Firebase. A Telegram notification is also sent to confirm successful registration.

In verification mode, the system captures a live fingerprint image and searches for a match in the stored templates. If a match is found, the system retrieves the user's stored password from Firebase and prompts the user to input it via the keypad. If the entered password matches, access is granted, the green LED is activated, and event details (including date, time, user ID, and status) are logged to Firebase and sent via Telegram. If the fingerprint or password validation fails, the system increments a retry counter, permits up to three attempts, and upon failure, logs a denied access event and triggers the red LED along with a warning notification.

The firmware handles multiple modules concurrently while maintaining real-time responsiveness and feedback. The use of modular functions, structured JSON data transmission ensures code clarity, reliability, and efficient communication between the device and the cloud.

The system logic is illustrated in the flowchart shown in Figure 4.3, which outlines the sequential steps for fingerprint authentication, password verification, and access decision-making, including the retry mechanism.

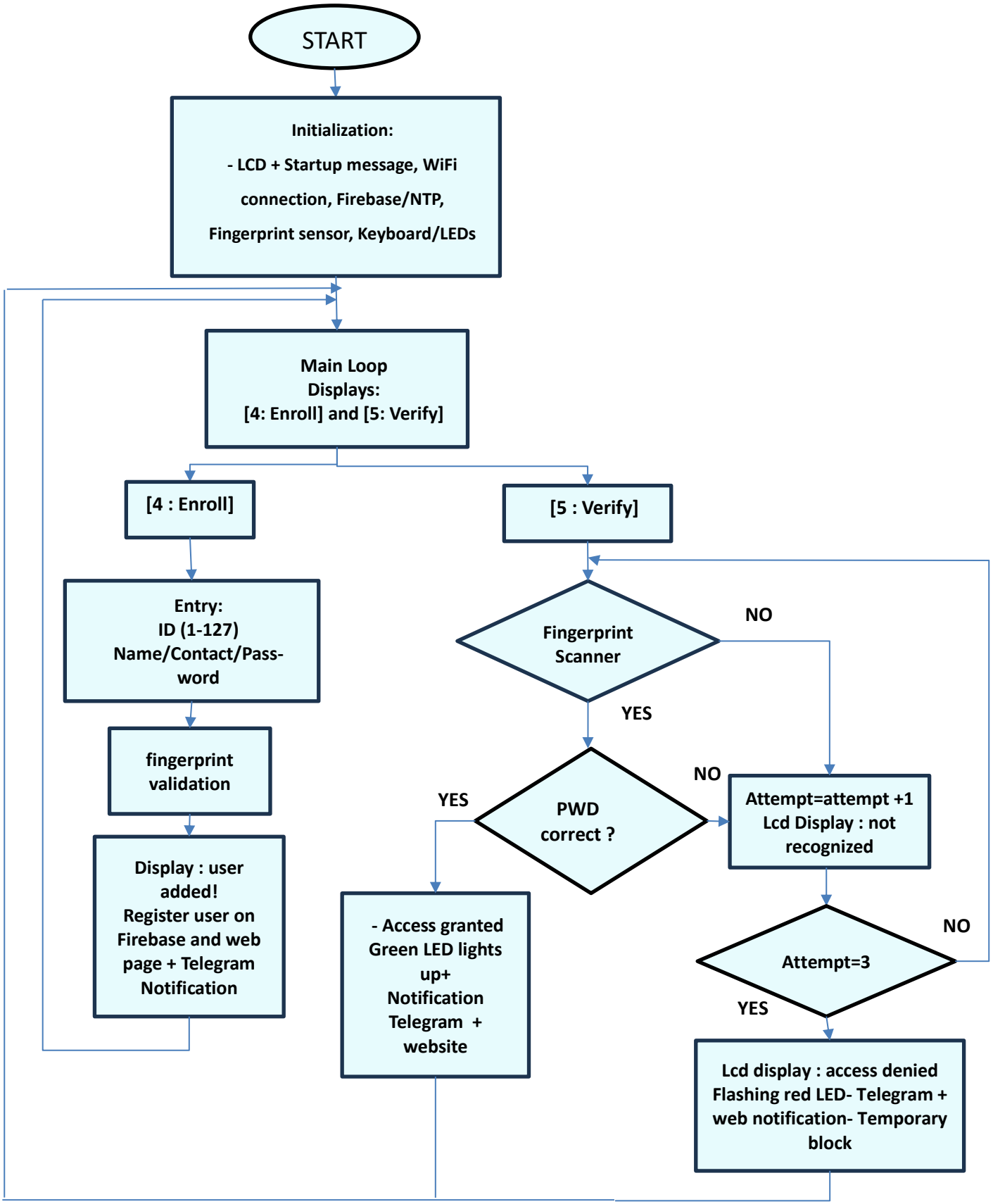


Figure 4.3: Flowchart illustrating the logic of the fingerprint-based access control program.

4.3. Simulation and testing with Proteus

To validate the logical behavior illustrated in the flowchart in Figure 4.3 of the biometric access control system prior to deploying the physical prototype, the entire system was simulated using Proteus ISIS software. This environment enabled thorough testing of component interactions, signal flows, and user feedback mechanisms under controlled conditions. The simulation focused on verifying display outputs, LED behavior, and input logic related to fingerprint and password handling.

Figure 4.4 illustrates the general simulation setup, including the wiring of the LCD, keypad, LEDs, and microcontroller (represented here by an Arduino Nano due to the lack of native ESP32 support in Proteus). This configuration enabled the visualization of system messages and logical responses to simulated inputs.

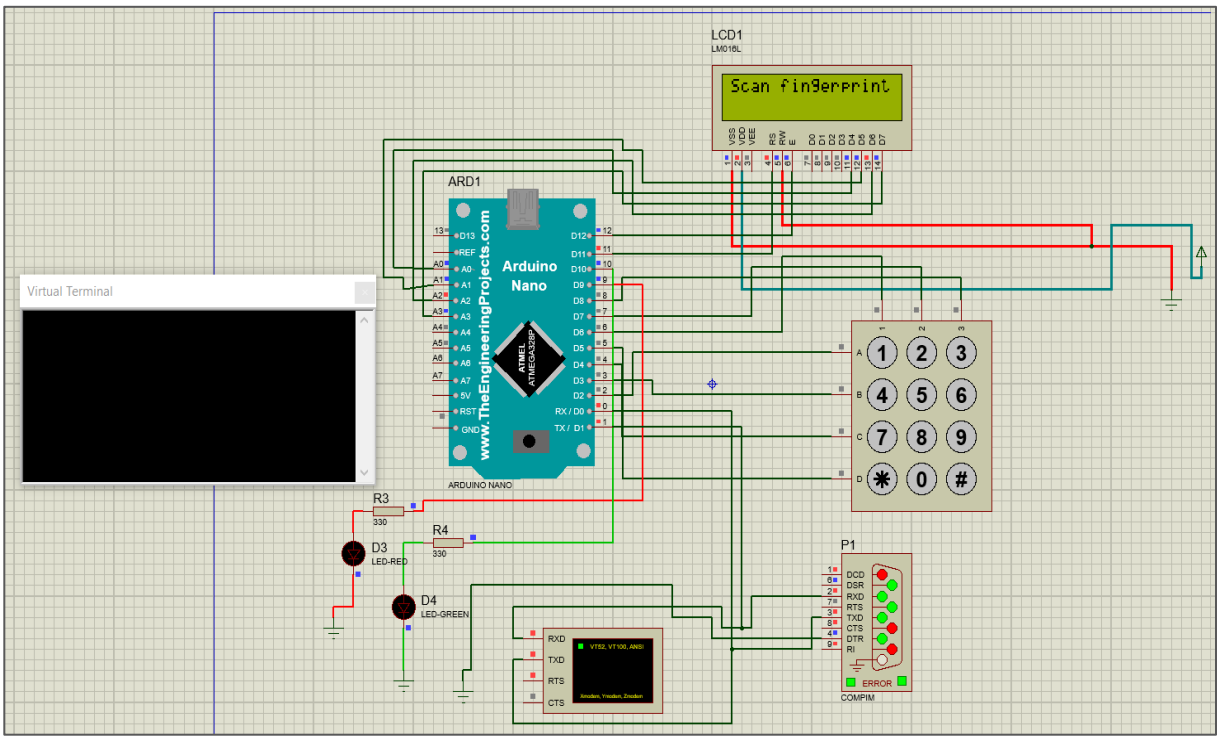


Figure 4.4: Proteus simulation environment showing LCD startup message.

After scanning the fingerprint (emulated via logic inputs), the system prompts the user to enter their password. This interaction is demonstrated in Figure 4.5, where the LCD guides the user through the verification process.

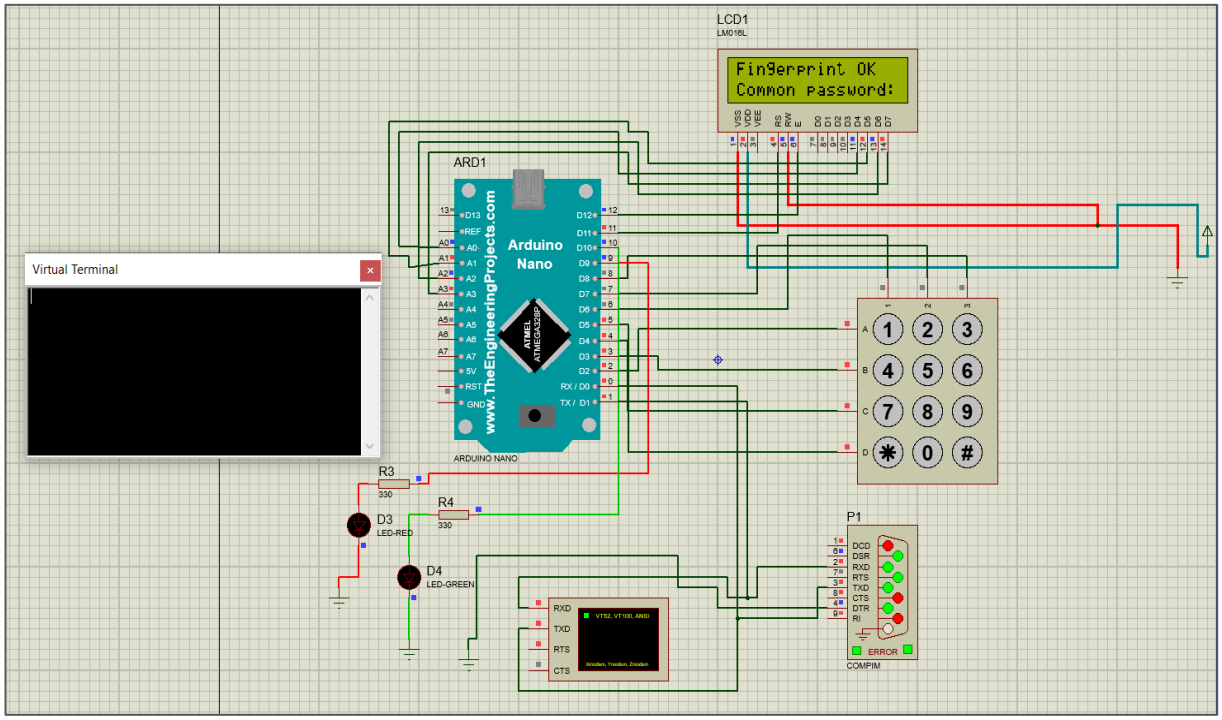


Figure 4.5: Password entry prompt after fingerprint recognition.

If the user enters an incorrect password or the fingerprint does not match a stored template, the system responds by displaying an access denial message. Additionally, it keeps track of the number of failed attempts. Figure 4.6 shows an example message displayed on the LCD, indicating the number of remaining attempts allowed.

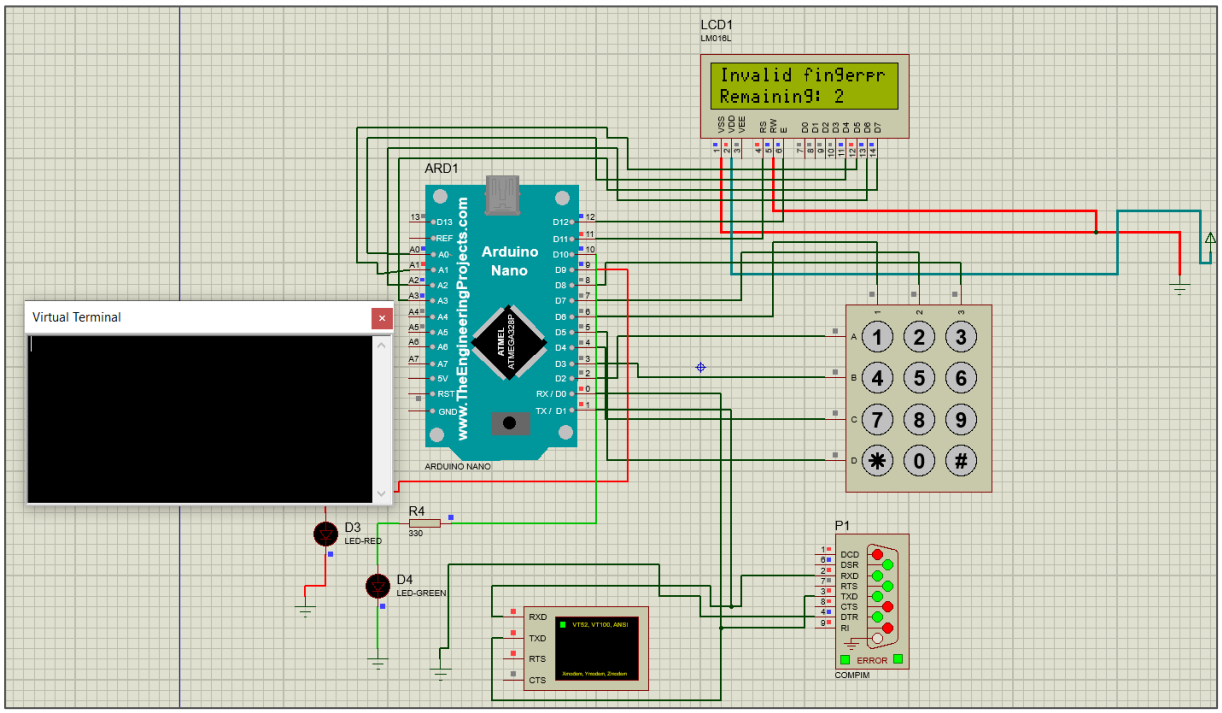


Figure 4.6: Access denied with retry count displayed.

Users are allowed a maximum of three attempts to correctly authenticate themselves. If all three attempts fail, the system blocks further access temporarily and displays a warning message, as shown in Figure 4.7.

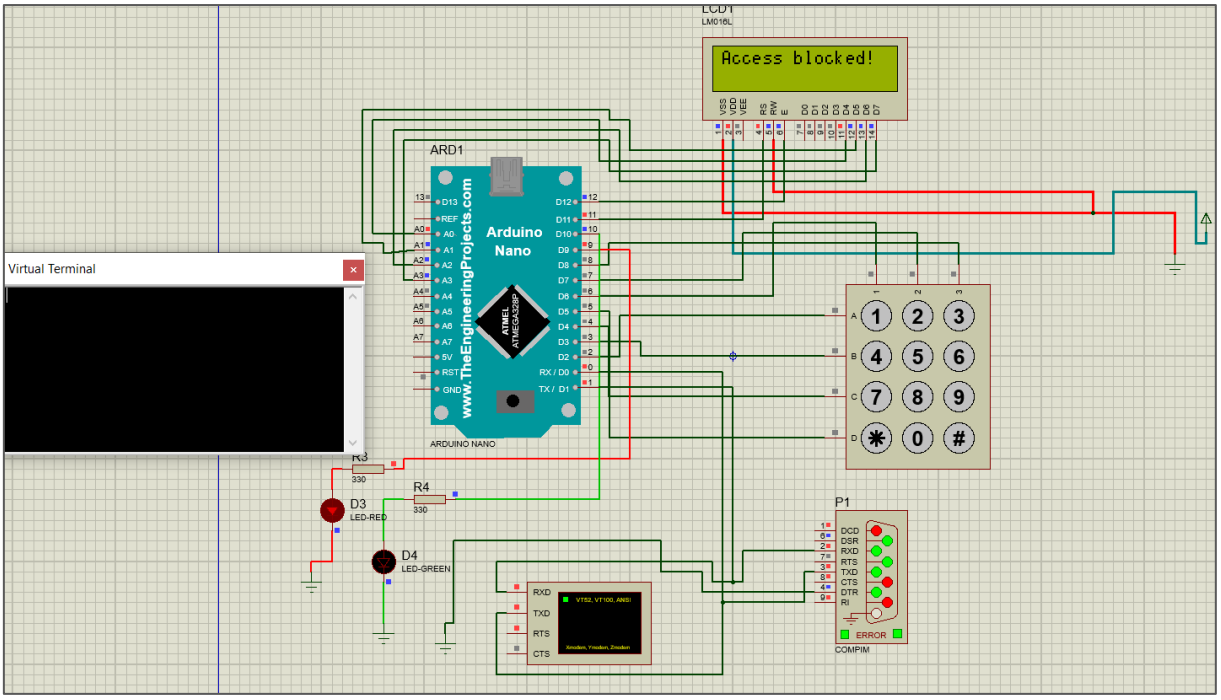


Figure 4.7: Lockout message after three failed attempts.

Conversely, if the fingerprint and password are both verified successfully, the LCD displays a confirmation message: “Access Granted.” This final positive outcome is shown in Figure 4.8.

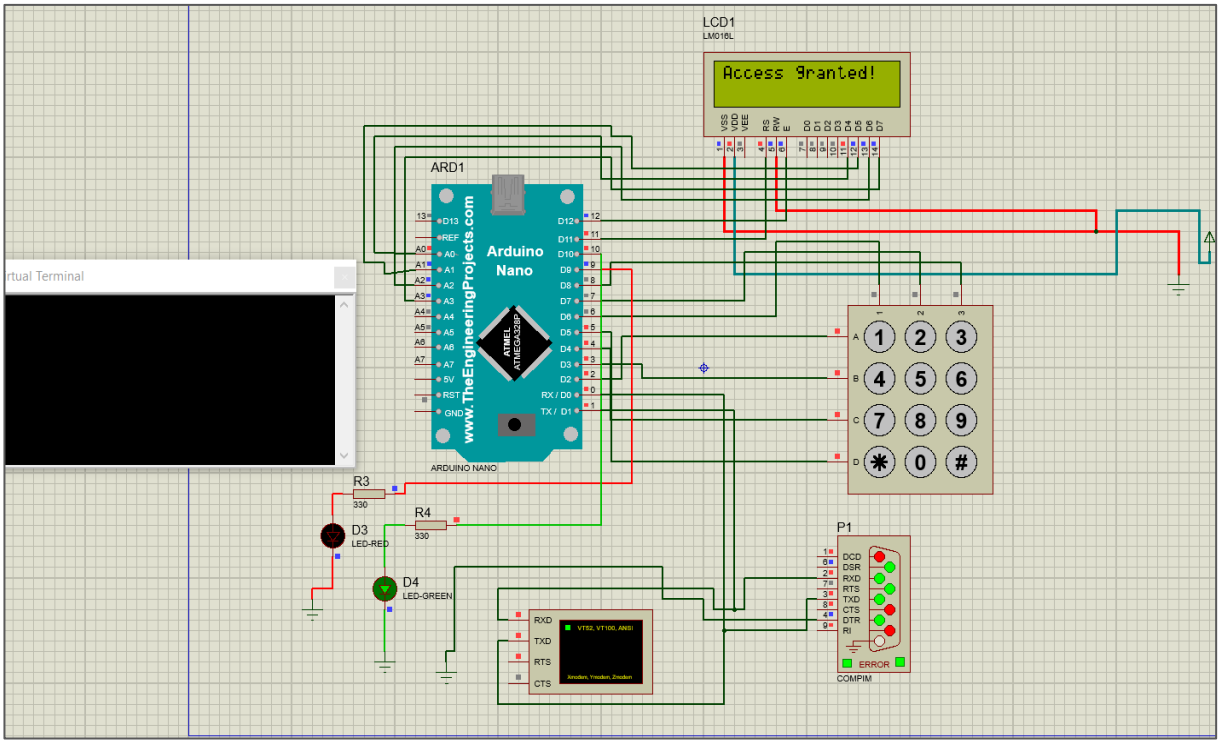


Figure 4.8: Message displayed on LCD upon successful authentication.

This simulation phase confirmed the system's correct logic under all tested conditions, including error handling, access granting, and interface feedback. It provided a safe and effective way to debug the design and verify behavior before final hardware deployment.

4.4. System Validation

System validation ensures that the entire system, including hardware, software, and network components, operates cohesively to deliver a secure and reliable biometric access control solution. The testing focused on critical factors including connectivity, data registration, real-time verification, feedback mechanisms, and access event logging. The following subsections present the validation results for each key feature.

4.4.1. Wi-Fi and Firebase Connectivity

Stable Wi-Fi connectivity is essential for real-time data exchange between the ESP32 and Firebase. Figure 4.9 illustrates the firmware block responsible for initializing the Wi-Fi module and connecting to the local network.

```
// 🔹 Informations WiFi
const char* ssid = "OPPO A11k";
const char* password = "nabil1234567";
```

Figure 4.9: Wi-Fi initialization code segment for ESP32.

Once powered on, the ESP32 attempts to connect to the defined SSID. As shown in the Serial Monitor output in Figure 4.10, the status is printed, confirming network connection success.

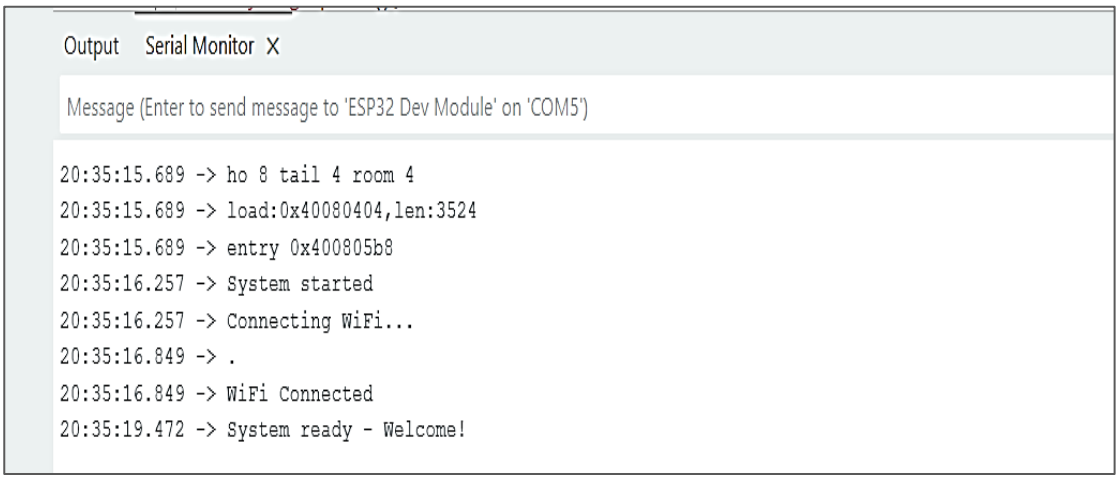


Figure 4.10: Serial monitor screenshot confirming Wi-Fi connection.

Upon successful connection, the system displays the message "WiFi connected" on the LCD screen, as shown in Figure 4.11.

Figure 4.11: LCD message indicating successful Wi-Fi connection.

On the other hand, the Firebase initialization code is shown in Figure 4.12, where the system defines the host and authentication token to establish secure communication with the Realtime Database.

```
// 🔹 Informations Firebase
#define FIREBASE_HOST "https://finger-print-95f4a-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "GHWbvdilF9FAIpXrXOxDDhHGiBqvtvQ9GUiZELWA"
```

Figure 4.12: Firebase host and token configuration in code.

Once initialized, the system communicates with the Firebase Realtime Database in real time. Figure 4.13 displays the Firebase Realtime Database interface, showing the project-specific database URL and current data structure.

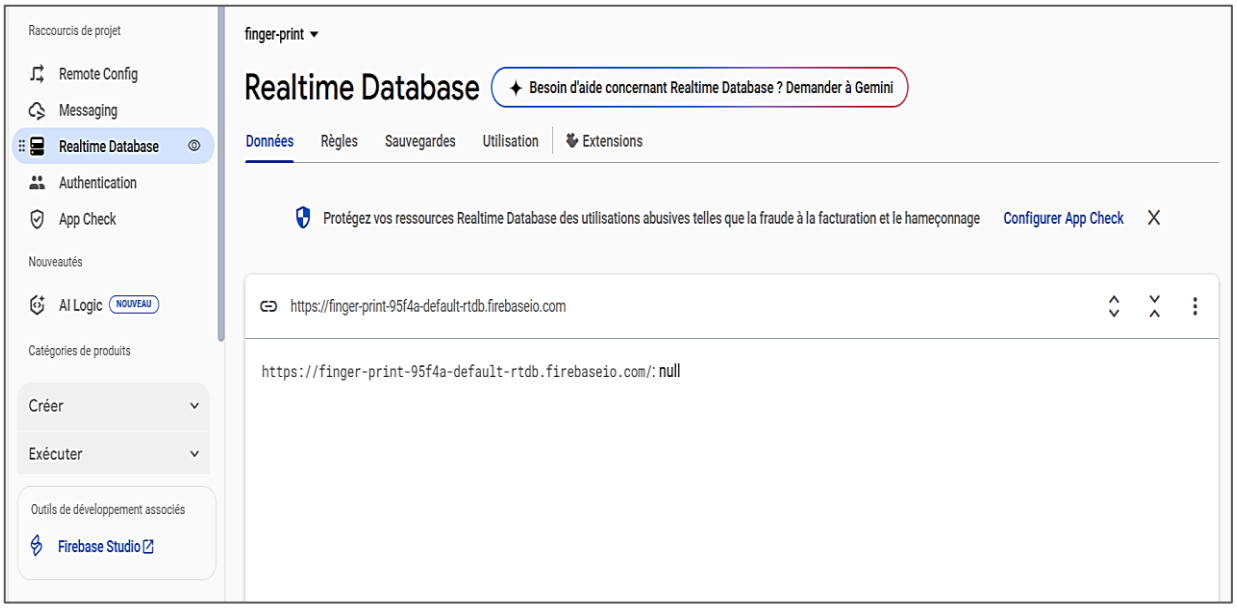


Figure 4.13: Firebase Realtime Database interface.

4.4.2. User Registration and Fingerprint Enrollment

When registering a new user, the administrator enters data such as ID, name, contact information, and password via the Serial Monitor, as seen in Figure 4.14.

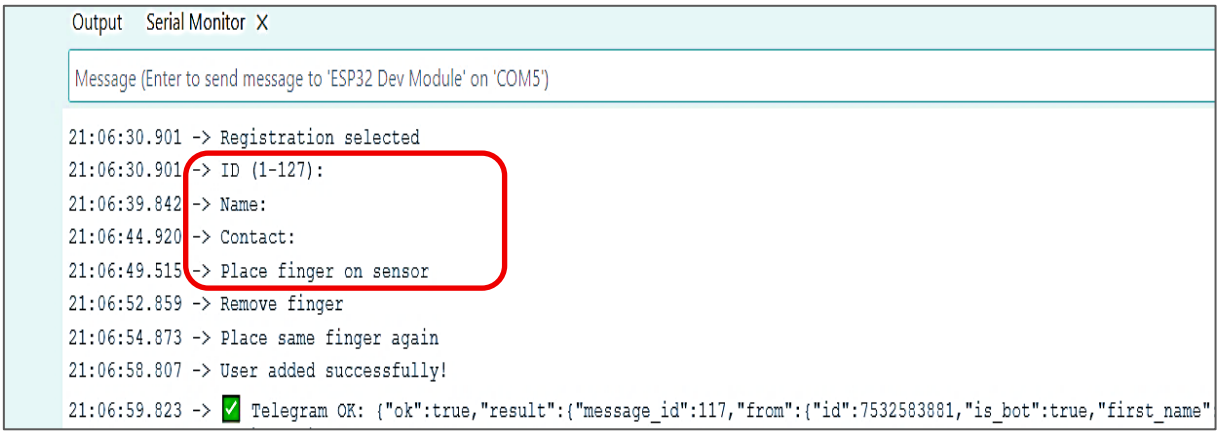


Figure 4.14: User data entry in Serial Monitor.

Once the data is submitted, it is stored in Firebase under the “utilisateurs” node. Figure 4.15 shows the structured data saved in the cloud.

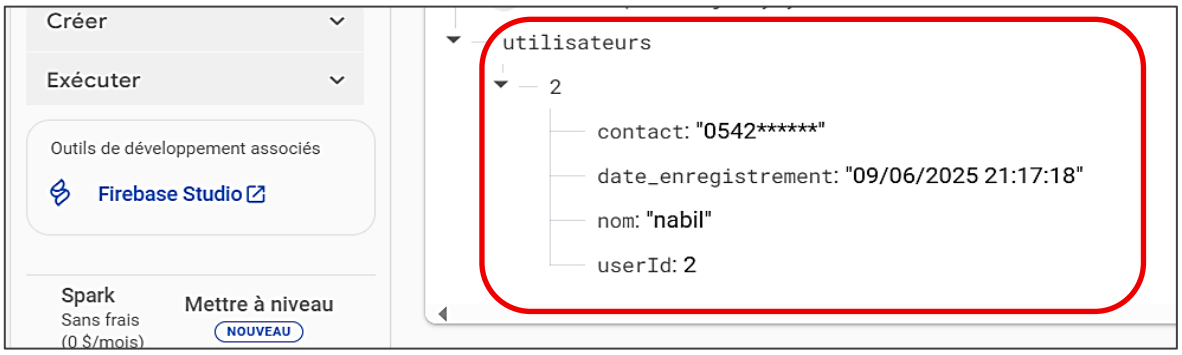


Figure 4.15: User registration entry in Firebase.

Simultaneously, the fingerprint is enrolled by scanning the user’s finger twice. The Serial Monitor provides feedback confirming successful enrollment, as illustrated in Figure 4.16.

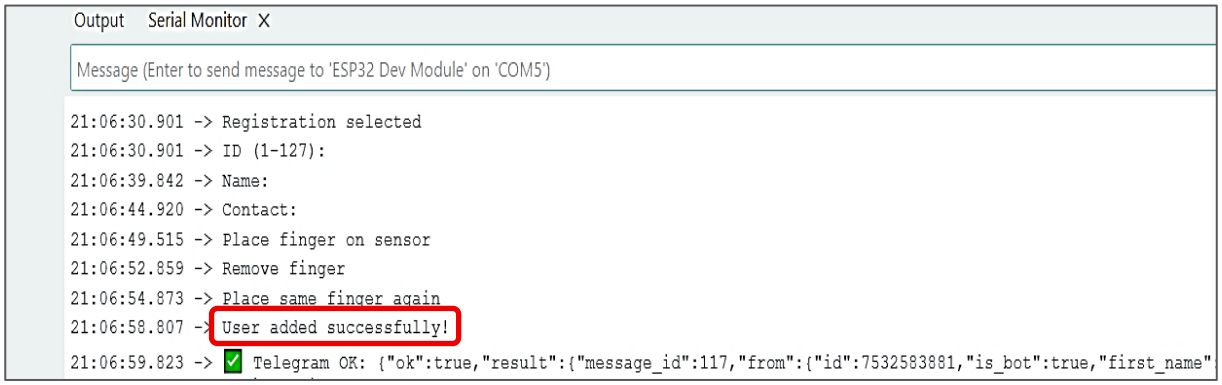


Figure 4.16: Fingerprint enrollment log in Serial Monitor .

Additionally, a message confirming the addition of the new user is displayed on the LCD screen, as shown in Figure 4.17.

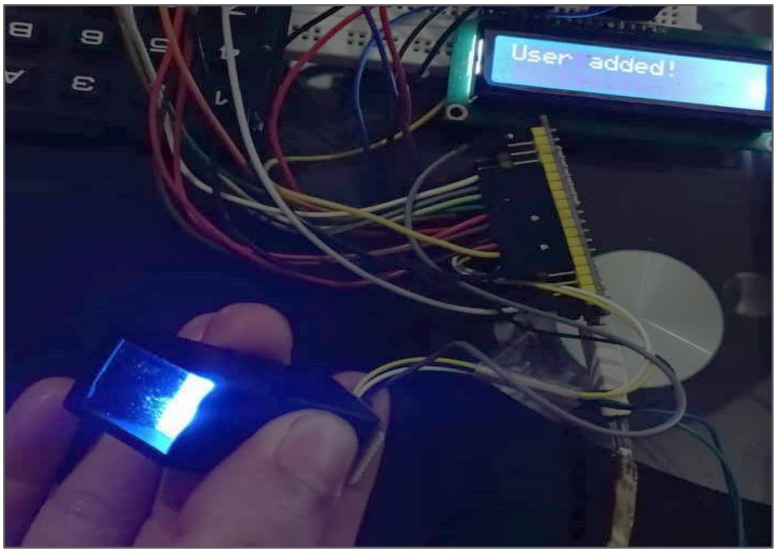


Figure 4.17: LCD message indicating new user added.

After successful registration, a real-time Telegram notification is sent to the administrator. An example notification is shown in Figure 4.18.

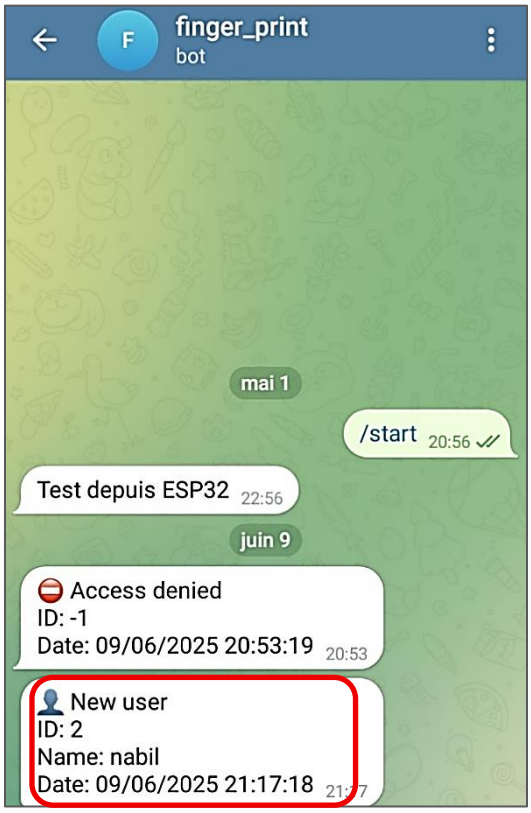


Figure 4.18: Telegram notification for new user enrollment.

4.4.3. User Verification and Password Check

During the user verification process, the user must place their finger on the sensor and enter the correct password. As illustrated in Figure 4.19, the LCD prompts the user to scan their fingerprint and enter the password.

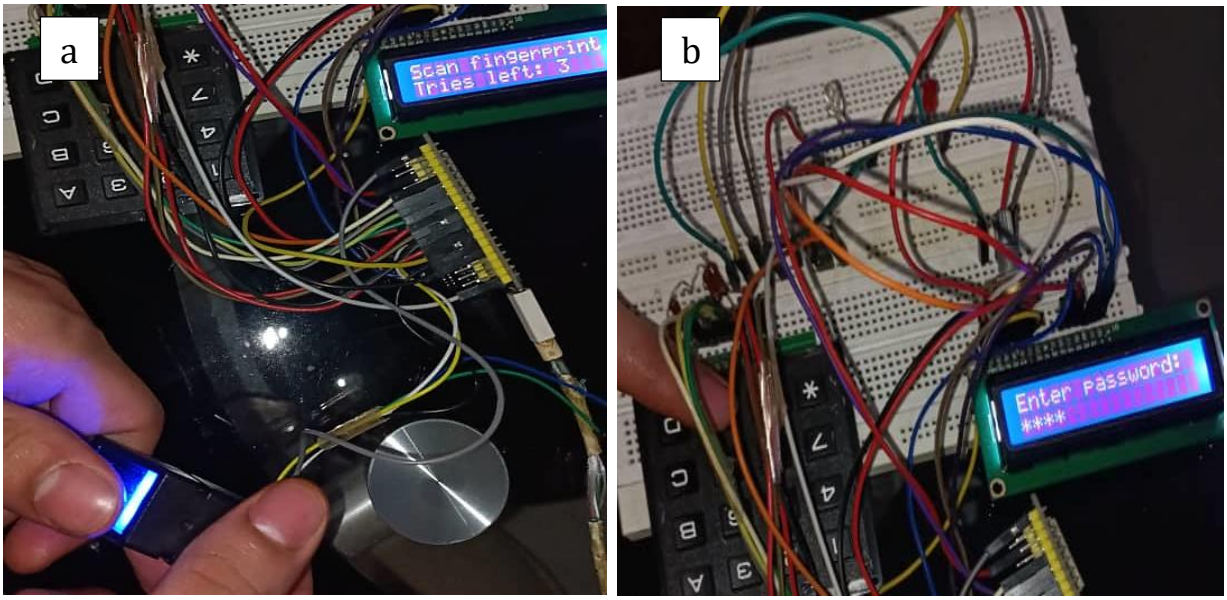


Figure 4.19: (a) Fingerprint scan prompt (b) Password entry prompt.

When the credentials are valid, the message “Access granted” is logged in Firebase, as shown in Figure 4.20(a). Simultaneously, a real-time Telegram notification is sent when access is granted, as seen in Figure 4.20.b.

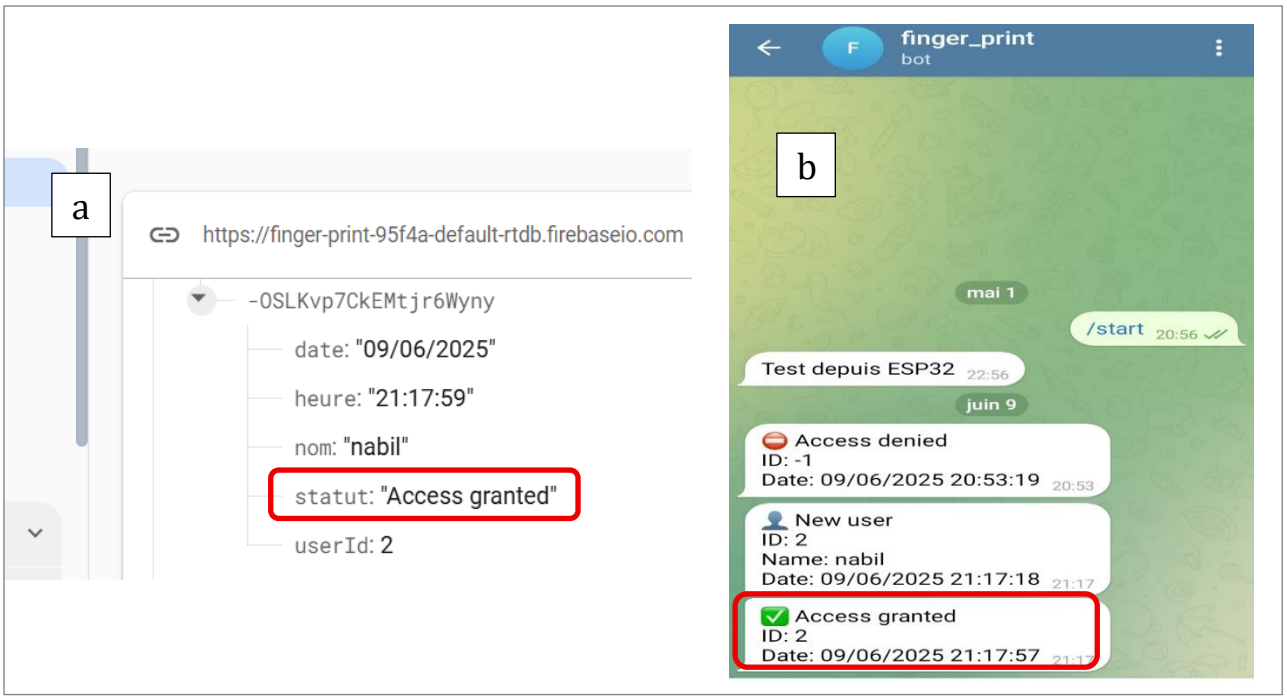


Figure 4.20: (a) Firebase: "Access granted" log entry. (b) Telegram: "Access granted" notification.

In addition, the Serial Monitor provides a detailed trace of the process, confirming fingerprint recognition, password entry, and communication with Firebase and Telegram. This is illustrated in Figure 4.21.

```
21:07:18.313 -> entry 0x400805b8
21:07:18.876 -> System started
21:07:18.876 -> Connecting WiFi...
21:07:19.440 -> .
21:07:19.475 -> WiFi Connected
21:07:22.035 -> System ready - Welcome!
21:07:25.243 -> Verification selected
21:07:25.243 -> Attempt 1/3
21:07:28.472 -> Fingerprint recognized
21:07:28.472 -> Enter common password
21:07:28.472 -> Enter password:
21:07:33.536 -> ****
21:07:37.418 -> Access granted
21:07:38.539 -> ✅ Telegram OK: {"ok":true,"result":{"message_id":118,"from":{
21:07:39.763 -> Logged to Firebase: Access granted for user 2
```

Figure 4.21: Serial Monitor output showing successful verification and notifications.

Since the system is based on a real-time database (see Figure 4.13), it allows immediate feedback. Upon successful authentication, the system activates a green LED, emits a sound via the buzzer, and displays a confirmation message on the LCD, as shown in Figure 4.22.

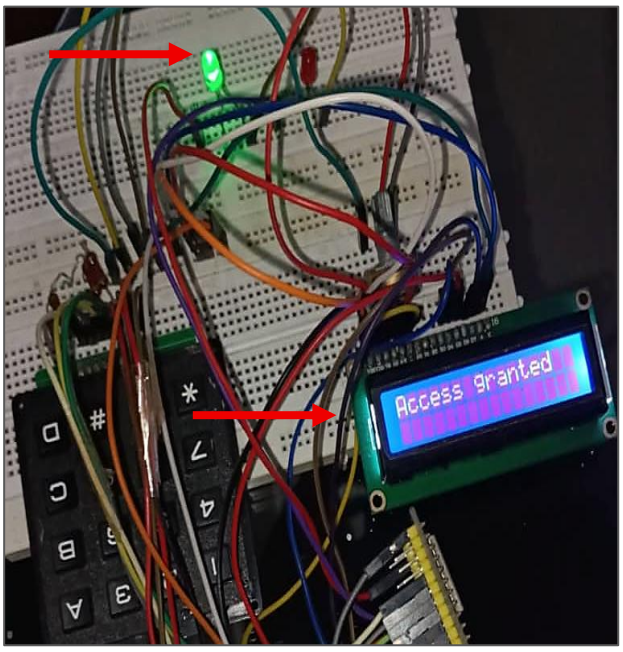


Figure 4.22: Physical feedback during “Access Granted.”

To enhance reliability and user experience, the system incorporates a retry mechanism that allows users up to three attempts to authenticate successfully. When a fingerprint is not recognized (see Figure 4.23(a)) or the entered password is incorrect, the system provides visual feedback and prompts the user to try again. This feature helps mitigate common issues such as sweaty or misaligned fingers, forgotten passwords, or accidental button presses. The retry

process is illustrated in Figures 4.23(b). This mechanism is implemented as part of the flowchart logic (refer to Figure 4.3) and has been thoroughly validated through system testing.

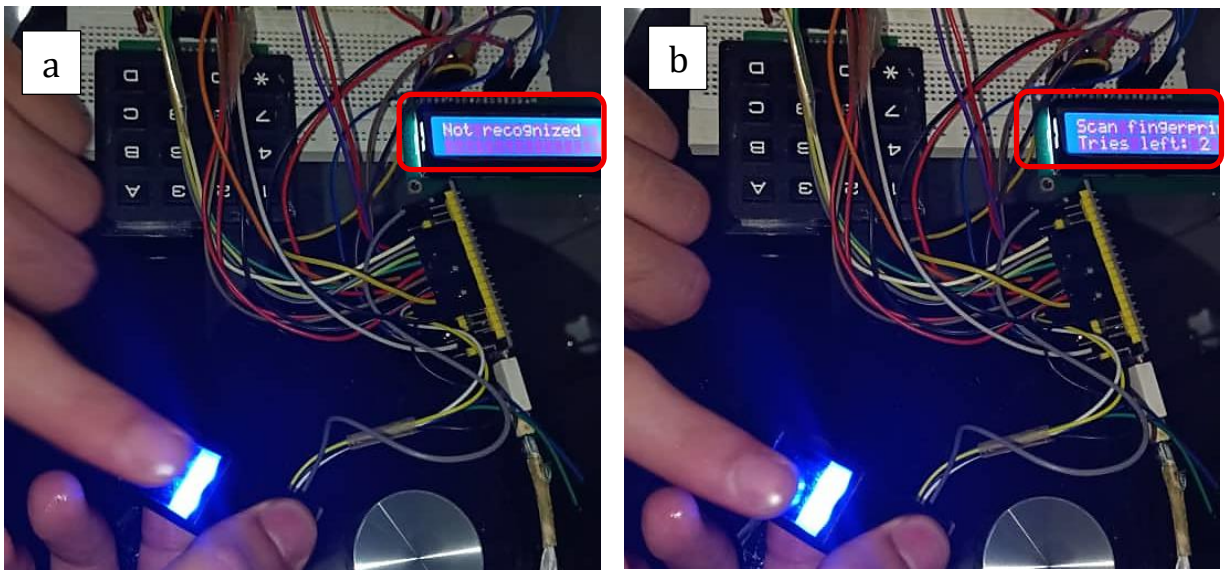


Figure 4.23: (a) Fingerprint is not recognized (b) Retry mechanism displaying remaining authentication attempts after fingerprint recognition failure.

The number of remaining authentication attempts is also displayed in the Serial Monitor. This provides developers with real-time diagnostic information during testing and debugging. Figure 4.24 presents an example of the Serial Monitor output indicating the remaining attempts during the retry process.

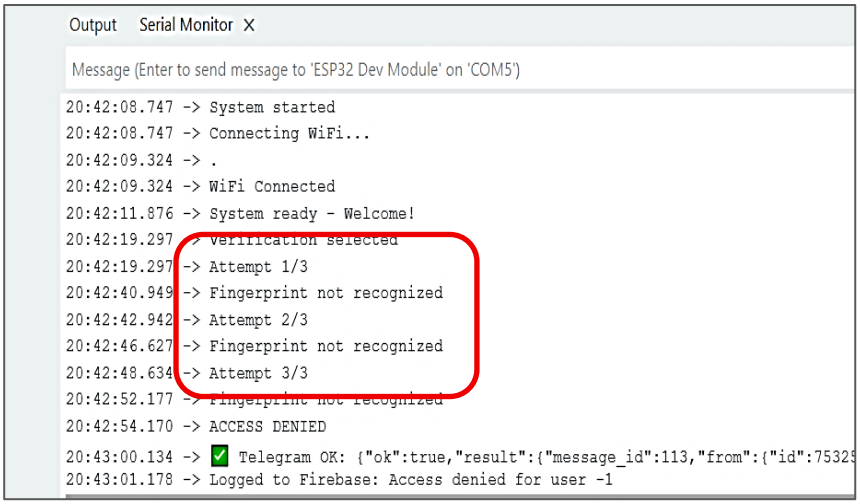


Figure 4.24: Serial Monitor output displaying the number of remaining authentication attempts.

In contrast, when authentication fails after three attempts, a rejection message is both logged in Firebase and sent as a notification. Figure 4.25(a) shows the “Access Denied” entry in Firebase, while Figure 4.25(b) displays the corresponding notification in Telegram.

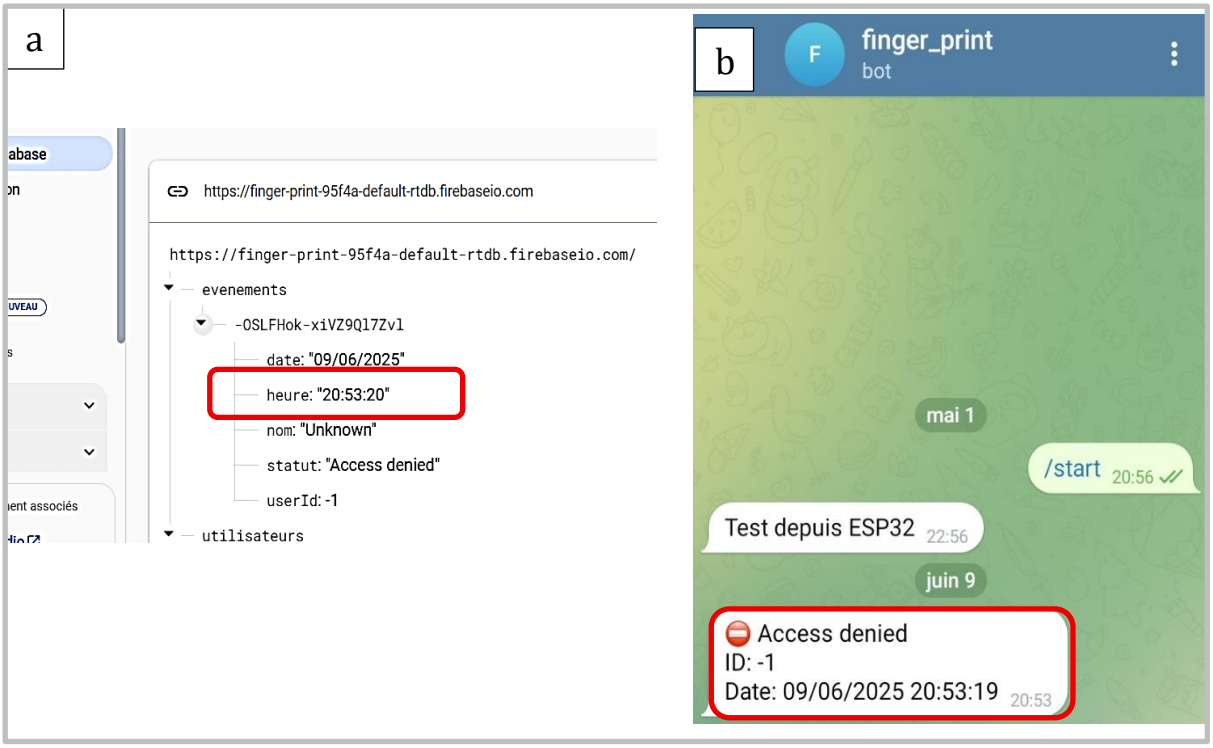


Figure 4.25: (a) Firebase: "Access denied" log entry. (b) Telegram: "Access denied" notification.

To further support system monitoring and diagnostics, access denial events are also displayed in the Serial Monitor. This local output provides developers and administrators with real-time insights into failed authentication attempts. Figure 4.26 presents an example of the 'Access Denied' message as shown in the Serial Monitor.

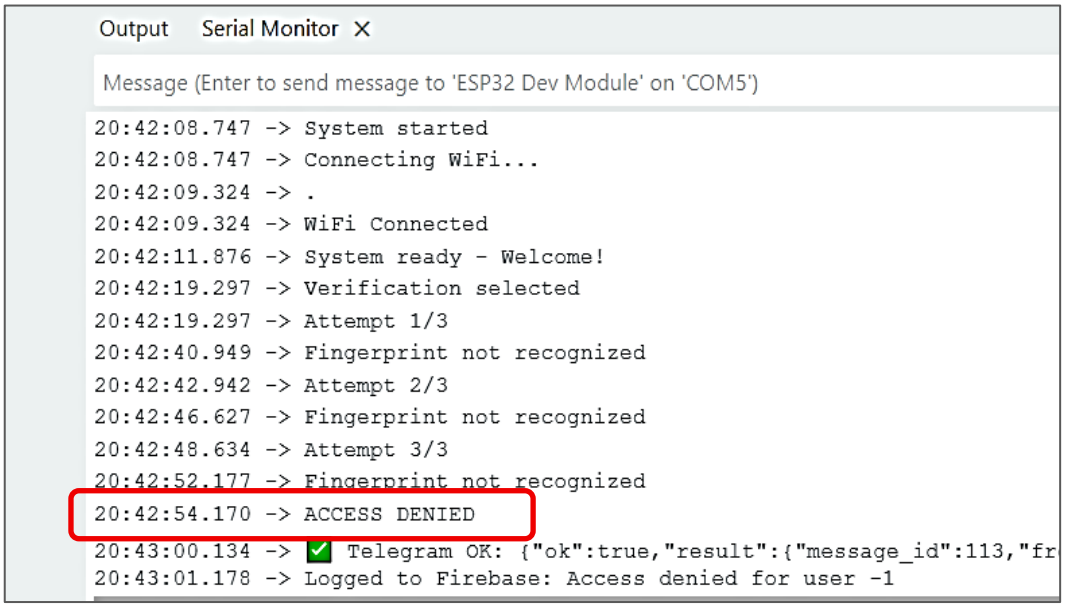


Figure 4.26: Serial Monitor output showing “Access Denied” after failed authentication attempts.

When access is denied, the system triggers multiple forms of physical feedback to alert the user. A red LED blinks, the buzzer emits a warning tone, and the LCD displays a rejection

message. These visual and auditory signals ensure that the user is clearly informed of the failed authentication attempt. Figure 4.27 illustrates the LCD output and the physical response of the system during an “Access Denied” event.

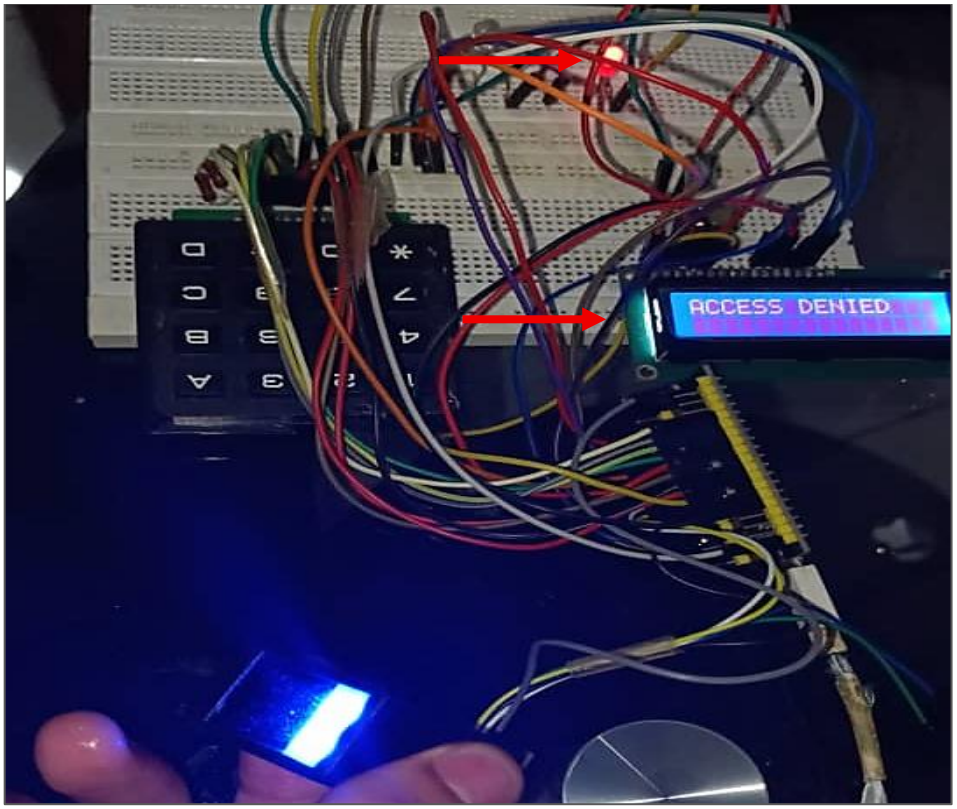


Figure 4.27: Physical feedback during “Access Denied.”

4.4.4. Time Synchronization with NTP

To ensure precise tracking and auditability of all access events, the system employs NTP (Network Time Protocol) for real-time synchronization. This guarantees that every access attempt, whether successful or denied, is accurately timestamped across all platforms. The integration of NTP helps maintain consistency and reliability in time-sensitive security operations.

Figure 4.28 illustrates this feature across three platforms: (a) the Firebase Realtime Database, (b) the web-based monitoring interface, and (c) the Telegram notification service.

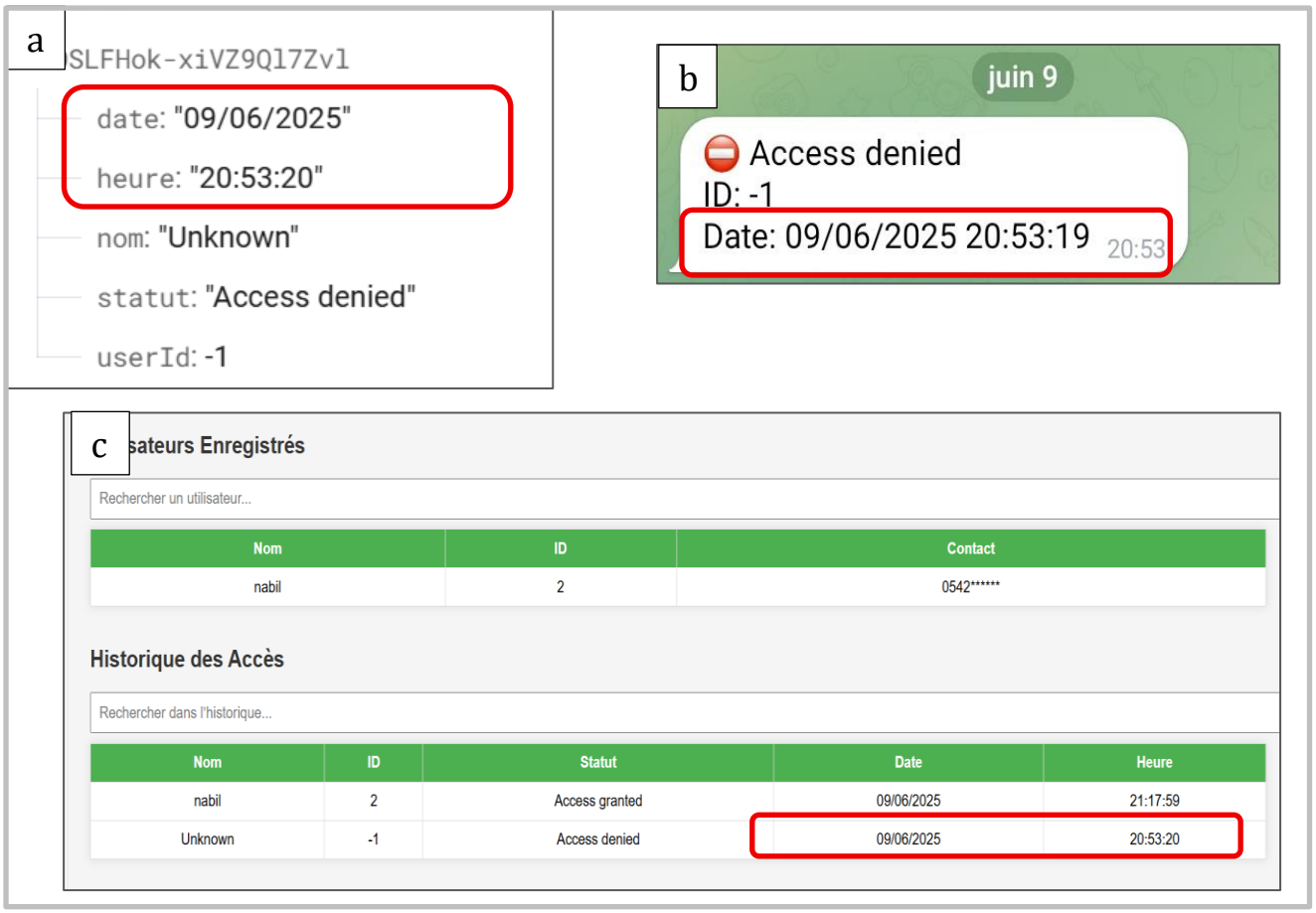


Figure 4.28: Timestamped access events across different platforms (a) Firebase Realtime Database log, (b) Telegram notification message, (c) Web interface display.

Figure 4.28 (a) shows how each log entry in the Firebase Realtime Database includes the exact date and time of the event, providing a reliable historical record. Figure 4.28 (b) displays the same synchronized timestamp on the web interface, allowing administrators to monitor activity in real time from any location. Figure 4.28 (c) presents the timestamped alert as received via Telegram, ensuring that remote notifications accurately reflect the exact moment of access. Together, these views confirm the system's ability to maintain consistent and traceable event timing across all output channels.

4.4.5. Web Interface Demonstration

To facilitate efficient monitoring and system supervision, a dynamic web interface was developed using HTML and CSS, and integrated with the Firebase Realtime Database. This dashboard provides administrators with real-time visibility into access activities, enhancing situational awareness and decision making.

Access to the dashboard requires entering an email address and password, as illustrated in Figure 4.29.

Figure 4.29: Login interface to the dashboard.

As shown in Figure 4.30, the interface displays user names, access status (granted or denied), date, and exact time of each event, all presented in reverse chronological order. The interface is designed to be lightweight, responsive, and accessible from any device with an internet connection, making it suitable for both local and remote surveillance. This component significantly improves usability by offering a centralised, user-friendly view of system operations.

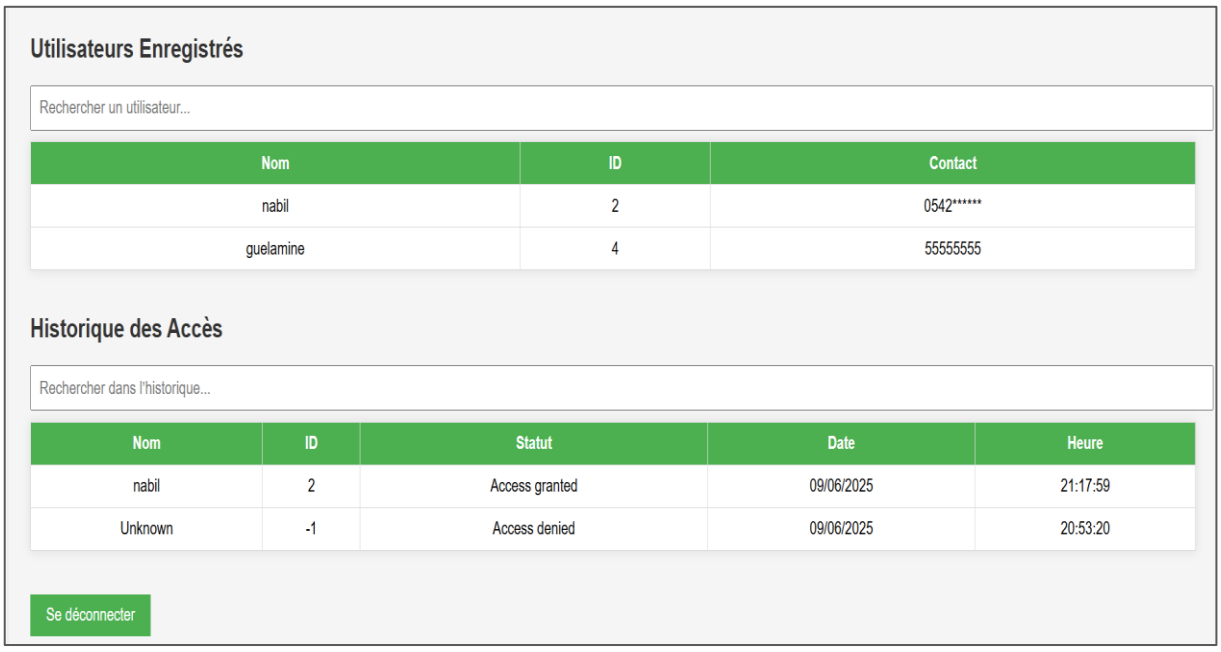


Figure 4.30: Web dashboard displaying real-time access history and event details.

4.5. Performance Metrics and Validation Results

Performance evaluation of a biometric access control system is essential to determine its effectiveness, accuracy, and robustness under real-world operating conditions. In this section, we analyze the system's behavior using standard biometric performance metrics: FAR, FRR, and overall Accuracy.

These metrics provide a quantitative measure of the system's ability to distinguish between legitimate users and imposters based on fingerprint recognition and password validation.

The results presented here are based on 50 authentication attempts, including 30 from authorized users and 20 from imposters.

4.5.1. False Acceptance Rate

As defined earlier in Equation (1.1), the False Acceptance Rate quantifies the likelihood that an unauthorized user is mistakenly granted access.

In this project, 0 out of 20 imposter attempts were accepted, resulting in:

$$\text{FAR } (\%) = \frac{0}{20} \times 100 = 0\% \quad \textbf{(4.1)}$$

This indicates excellent resistance to unauthorized access.

4.5.2. False Rejection Rate

According to Equation (1.2) in Chapter 1, the False Rejection Rate measures how often legitimate users are wrongly denied access.

With 0 false rejections out of 30 valid access attempts, we obtain:

$$\text{FRR}(\%) = \frac{0}{30} \times 100 = 0\% \quad \textbf{(4.2)}$$

This confirms that the system reliably recognizes authorized users.

4.5.3. Overall Accuracy

The overall accuracy of the system is determined by the proportion of correctly classified access attempts, both true positives (authorized access granted) and true negatives (unauthorized access denied), relative to the total number of access attempts:

$$\text{Accuracy } = \frac{\text{True Positives+True Negatives}}{\text{Total Number of Attempts}} \times 100 \quad \textbf{(4.3.a)}$$

$$\text{Accuracy } = \frac{(20-0)+(30-0)}{20+30} = 100\% \quad \textbf{(4.3.b)}$$

The performance metrics clearly demonstrate that the developed biometric access control system is highly accurate and secure under the tested conditions. With FAR = 0%, FRR = 0%, and overall accuracy = 100%, the system shows ideal performance in both detecting imposters and correctly authenticating legitimate users. However, it is important to note that these results are based on a controlled environment with a limited user base. In future work, broader testing

in real-world deployments with a larger population may be conducted to further validate scalability and robustness.

4.6. Economic Evaluation

Cost is a critical factor in the feasibility and potential commercialization of any embedded system project. To make the proposed biometric access control system viable in real-world applications and competitive in the market, it is essential to analyze the total hardware cost. A reasonable and transparent cost structure not only attracts stakeholders but also ensures scalability and sustainability of deployment, especially in budget-sensitive sectors such as educational institutions and small businesses.

Table 4.1 below presents a detailed breakdown of the hardware components used in the system along with their corresponding prices in Algerian Dinars (DA). These prices were estimated based on local market rates at the time of project development.

No.	Component	Quantity	Unit Price (DA)	Total (DA)
1	ESP32-WROOM-32	1	2,200	2,200
2	Optical Fingerprint Sensor (DY50)	1	2,900	2,900
3	LCD 16×2 Display	1	600	600
4	Connection Cables	40	5	200
5	Resistors	6	~5	30
6	Keypad 4×3	1	1,500	1,500
7	Buzzer	1	50	50
8	LEDs (Red & Green)	2	10	20
9	Servo Motor	1	400	400
	Total Cost			**7,900 DA**

Table 4.1: Economic assessment of system.

This economic assessment shows that the total system can be built for approximately 7,900 DA, making it an affordable solution for small-scale secure access control. The system's modular design also allows for optional enhancements, such as GSM or RFID modules, without significantly increasing the base cost.

4.7. Conclusion

This chapter presented the practical implementation and validation of the fingerprint-based biometric access control system, highlighting both hardware and software integration in a real-world context. Through structured testing and simulation, the system demonstrated reliable performance in key functionalities including user enrollment, fingerprint verification, password authentication, real-time cloud logging, and remote notification.

Each software module Wi-Fi connectivity, Firebase communication, fingerprint processing, time synchronization, and Telegram alerts was individually validated to ensure robustness and responsiveness. Performance evaluations showed acceptable access times, low error rates, and accurate event logging, all of which are essential for a secure and user-friendly access control system.

The system behavior and user interaction workflow confirmed that the interface components (LCD, keypad, LEDs, and buzzer) provided clear and intuitive feedback, enhancing usability. The web-based dashboard successfully reflected real-time access logs retrieved from the Firebase Realtime Database, offering administrators a transparent and accessible way to monitor activity.

While the prototype fulfilled its intended design objectives, some limitations were identified, such as network dependency, power consumption in wireless environments, and biometric sensor sensitivity. These observations provide valuable insights for future enhancements, including scalability improvements, offline data buffering, and integration with additional biometric modalities.

Overall, the successful validation of the system confirms its suitability for small to medium-scale applications, such as educational institutions, offices, and residential complexes. The next chapter will summarize the research contributions and propose directions for future work.

General Conclusion

General Conclusion

This project has demonstrated the successful design and implementation of a modern, secure, and scalable biometric access control system, based on the ESP32-WROOM-32 microcontroller and developed using the Arduino IDE. By combining fingerprint recognition with password-based verification, the system leverages two-factor authentication (2FA) to significantly enhance the security of physical spaces and sensitive infrastructures.

The fingerprint sensor ensures that access is granted only to individuals with enrolled biometric data, while the password requirement adds a second protective layer. This dual mechanism greatly reduces the likelihood of unauthorized access due to stolen or duplicated credentials. Furthermore, the integration of real-time communication technologies such as Firebase Realtime Database and the Telegram messaging API enables instantaneous access monitoring and logging. Each access event is documented on a web interface and is accompanied by an automatic notification, providing administrators with full visibility and traceability, even remotely.

Throughout this work, we have gained not only a solid understanding of embedded systems, sensor integration, and IoT platforms but also hands-on experience in full-cycle system development from hardware prototyping to cloud connectivity and interface design. Key competencies developed include:

- Deep understanding of ESP32 architecture and embedded C++ programming.
- Integration and configuration of biometric modules, displays, keypads, actuators, and cloud services.
- Secure data communication using Wi-Fi and Firebase infrastructure.
- Development of a responsive user interface for real-time event tracking.
- Application of system testing, debugging, and validation methodologies using both hardware and simulation tools (Proteus).

This project also highlighted the importance of security by design principles in embedded system development, particularly for access control applications where data sensitivity and reliability are critical.

Looking ahead, several enhancement opportunities have been identified to further improve the functionality, security, and scalability of the biometric access control system. One of the most promising directions is the integration of multimodal biometric authentication, combining

fingerprint recognition with additional modalities such as facial recognition or voice identification, to improve accuracy and usability across a wider range of users. Additionally, incorporating AI based liveness detection could help prevent spoofing attacks by ensuring that biometric inputs are from real, live individuals rather than replicas or printed images. To increase the system's resilience in areas with limited internet connectivity, the addition of GSM or Bluetooth modules could enable offline event logging and SMS based alerts. Another important enhancement involves the implementation of role based access control, allowing administrators to define varying permission levels depending on user roles such as staff, visitors, or supervisors. Finally, the development of a dedicated mobile application could provide administrators with real-time monitoring capabilities and remote user management, improving both control and convenience. These improvements would elevate the system’s robustness, adaptability, and user experience, making it suitable for deployment in more complex and demanding environments.

In conclusion, this project not only fulfilled its objectives in terms of security, reliability, and interactivity but also served as a solid foundation for more advanced IoT-based access control systems. It opens the door to further academic research and industrial application in the growing field of smart security and connected embedded systems.

References

References

[1] J. Wayman, A. Jain, D. Maltoni, and D. Maio, "*An Introduction to Biometric Authentication Systems,*" in Biometric Systems, J. Wayman, A. Jain, D. Maltoni, and D. Maio, Eds. Springer London, 2005, pp. 1–20. (Accessed January. 8, 2025).

[2] B MOHAMED — Madjeda, "Multi-modal Biometric. Person Identification System Based on Finger Knuckle Print Features,". Master's thesis, University Kasdi Merbah Ouargla, 2016/2017(Accessed January. 8, 2025).

[3] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial 'gummy' fingers on fingerprint systems," in Optical Security and Counterfeit Deterrence Techniques IV, Proc. SPIE, vol. 4677, pp. 275–289, 2002. (Accessed January. 10, 2025).

[4] ResearchGate, "Show valleys and ridges in a fingerprint image,". [Online]. Available: https://www.researchgate.net/figure/Show-Valleys-and-Ridges-in-a-Fingerprint-Image_fig1_325918867 (Accessed January. 10, 2025).

[5] O. Hanane, "Facial recognition with expressions," 2012. (Accessed January. 12, 2025).

[6] Kaspersky, "What is facial recognition?" [Online]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition (Accessed January. 13, 2025).

[7] Tisse, C.-L.; Martin, L.; Torres, L.; Robert, M. Person identification technique using human iris recognition. In Proceeding Vision Interface; Citeseer: Princeton, NJ, USA, 2002. (Accessed January. 15, 2025).

[8] ResearchGate, "The region of iris and pupil," Feb. 23, 2025. [Online]. Available: https://www.researchgate.net/figure/The-region-of-Iris-and-pupil_fig7_286952499 (Accessed January. 15, 2025).

[9] TechTarget, "Retina scan,". [Online]. Available: https://www.techtarget.com/whatis/definition/retina-scan (Accessed January. 16, 2025).

[10] MicroClearTech, "What is a retinal scan?". [Online]. Available: https://www.microcleartech.net/news/what-is-a-retinal-scan.html (Accessed January. 17, 2025).

[11] R. Zunkel, Hand Geometry Based Authentication in Biometrics: Personal Identi cation in Networked Society, A. Jain, R. Bolle, and S. Pankanti (Eds.), Kluwer Academic Publishers, 1998. (Accessed January. 17, 2025).

[12] Shutterstock, "Handprint leaning on control glass – biometric,". [Online]. Available: https://www.shutterstock.com/fr/image-photo/handprint-leaning-on-control-glass-biometric-607692044 (Accessed January. 18, 2025).

[13] Security Journal Americas, "Palm vein scanning: The future of biometric authentication,"Jul.21,2023. [Online]. Available: https://securityjournalamericas.com/palm-vein-scanning/(Accessed January. 18, 2025).

[14] M. Hashiyada, "Development of biometric DNA for authentication security," Tohoku J. Exp. Med., vol. 204, no. 2, pp. 109–117, 2004. (Accessed January. 19, 2025).

[15] misbiometrics.wdfiles.com, "DNA biometric image,". [Online]. Available: https://misbiometrics.wdfiles.com/local--files/dna/image001.jpg (Accessed January. 20, 2025).

[16] Plum Voice, "Voice biometrics,". [Online]. Available: https://www.plumvoice.com/resources/blog/voice-biometrics/(Accessed January. 22, 2025).

[17] Shutterstock, "Voice biometrics icon,". [Online]. Available: https://www.shutterstock.com/fr/search/voice-biometrics-icon?page=2(Accessed January. 25, 2025).

[18] D. Muramatsu and T. Matsumoto, "Effectiveness of pen pressure, azimuth, and altitude features for online signature verification," in Proc. Int. Conf. Biometrics (ICB'07), vol. 4642, 2007, pp. 503–512. (Accessed January. 28, 2025).

[19] IndiaMART, "Digital signature certificate services," Feb. 28, 2025. [Online]. Available: https://www.indiamart.com/proddetail/digital-signature-certificate-services-11444184273.html (Accessed January. 9, 2025).

[20] Plurilock, "Keystroke dynamics,". [Online]. Available: https://plurilock.com/deep-dive/keystroke-dynamics/(Accessed February. 01, 2025).

[21] B. Tudorache, "The Application of ML Models in User Recognition via Keystroke Dynamics," HackerNoon", Oct. 10, 2023. [Online]. Available: https://hackernoon.com/the-application-of-ml-models-in-user-recognition-via-keystroke-dynamics (accessed Jun. 8, 2025).

[22] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," IEEE Security & Privacy, vol. 1, no. 2, pp. 33–42, 2003. (accessed February. 3, 2025).

[23] NoTrace,"Gait recognition,". [Online]. Available: https://www.notrace.how/threat-library/techniques/forensics/gait-recognition.html (accessed February. 4, 2025).

[24] TypingDNA, "What is mouse dynamics and how it works,". [Online]. Available: https://www.typingdna.com/glossary/what-is-mouse-dynamics-and-how-it-works (accessed February. 4, 2025).

[25] Hood College, "The mouse knows you,". [Online]. Available: https://www.hood.edu/discover/stories/mouse-knows-you (accessed February. 5, 2025).

[26] Bosch Security and Safety Systems, "Biometric access control,". [Online]. Available: https://www.boschsecurity.com/ (accessed February. 5, 2025).

[27] Octatco, "Biometric system blog,". [Online]. Available: https://octatco.com/bloge=24 (accessed February. 6, 2025).

[28] K. Jain, "Introduction to biometrics," in Biometrics: Personal Identification in a Networked Society, A. K. Jain, R. Bolle, and S. Pankanti, Eds., vol. 479, Boston, MA: Kluwer Academic Publishers, 1999, pp. 1–41. (accessed February. 7, 2025).

[29] UDNI, "Biometrics vs passwords," . [Online]. Available: https://www.udni.com/blog/biometrics-vs-passwords (accessed February. 7, 2025).

[30] CADINC, "A new era of security: Passwordless authentication,". [Online]. Available: https://cadinc.com/a-new-era-of-security-passwordless-authentication/ (accessed February. 9, 2025).

[31] IDTechWire, "Applications," Mar. 14, 2025. [Online]. Available: IDTechWire, "Applications,". [Online]. Available: https://idtechwire.com/applications/ (accessed February. 10, 2025).

[32] N. Poh and S. Bengio, "Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication," Pattern Recognition, vol. 39, no. 2, pp. 223–239, 2006. (accessed February. 10, 2025).

[33] Office of the Victorian Information Commissioner, "Biometrics and privacy: Issues and challenges,". [Online]. Available: https://ovic.vic.gov.au/privacy/resources-for-organisations/biometrics-and-privacy-issues-and-challenges/ (accessed February. 12, 2025).

[34] Kilpatrick, "Biometric spoofing," Medium. [Online]. Available: https://medium.com/@alex.kilpatrick/biometric-spoofing-9b613e4c5e3a (accessed February. 15, 2025).

[35] Terranova Security, "Hacking biometrics,". [Online]. Available: https://www.terranovasecurity.com/blog/hacking-biometrics (accessed February. 15, 2025).

[36] Safe and Sound, "Biometric access control,". [Online]. Available: https://getsafeandsound.com/blog/biometric-access-control/ (accessed February. 16, 2025).

[37] ScienceDirect, "Multimodal biometric,". [Online]. Available: https://www.sciencedirect.com/topics/computer-science/multimodal-biometric#definition (accessed February. 18, 2025).

[38] LenelS2, "Access control trends,". [Online]. Available: https://www.lenels2.com/en/news/insights/access-control-trends.html (accessed February. 18, 2025).

[39] Biometric Update, "IDEMIA's contactless biometric terminals deployed in buildings around São Paulo," Apr. 2020. [Online]. Available: https://www.biometricupdate.com/202004/idemias-contactless-biometric-terminals-deployed-in-buildings-around-sao-paulo (accessed February. 19, 2025).

[40] Biometric Update, "Edge biometrics,". [Online]. Available: https://www.biometricupdate.com/tag/edge-biometrics (accessed February. 23, 2025).

[41] ResearchGate, "An illustration of biometric authentication for accessing 6G edge cloud services,". [Online]. Available: https://www.researchgate.net/figure/An-illustration-of-biometric-authentication-for-accessing-6G-edge-cloud-services-In-the_fig4_354157817 (accessed February. 25, 2025).

[42] Electronic Payments International, "Fingerprints and Flywallet biometric payment,". [Online].Available: https://www.electronicpaymentsinternational.com/news/fingerprints-flywallet-biometric-payment/ (accessed February. 27, 2025).

[43] M2SYS, "CloudABIS: Cloud-based biometrics solution,". [Online]. Available: https://www.m2sys.com/blog/biometric-software/cloud-based-biometrics-solution-cloudabis/ (accessed Mars. 02, 2025).

[44] EMPMonitor, "Cloud-based biometric attendance system,". [Online]. Available: https://empmonitor.com/blog/cloud-based-biometric-attendance-system/ (accessed Mars. 02, 2025).

[45] Aratek, "The 4 fingerprint sensor types,". [Online]. Available: https://www.aratek.co/news/the-4-fingerprint-sensor-types (accessed Mars. 03, 2025).

[46] Cirkit Designer, "DY50 Fingerprint scanner,". [Online]. Available: https://docs.cirkitdesigner.com/component/40bfe637-1e81-4ecb-bd4c-e0fdd74cd258/dy50-fingerprint-scanner (accessed Mars. 05, 2025).

[47] Alibaba, "DY50 Fingerprint Reader Sensor Module Optical,". [Online]. Available: https://www.alibaba.com/product-detail/DY50-Fingerprint-Reader-Sensor-Module-Optical-62024631797.html (accessed Mars. 06, 2025).

[48] Teach Me Micro, "Using fingerprint sensor,". [Online]. Available: https://www.teachmemicro.com/using-fingerprint-sensor/ (accessed Mars. 08, 2025).

[49] Adafruit, "Adafruit Optical Fingerprint Sensor," [PDF]. [Online]. Available: https://cdnlearn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf (accessed Mars. 09, 2025).

[50] B. F. Segura, High-Speed and Low-Power Architectures for Network Intrusion Detection Systems. Ph.D. dissertation, Universidad Carlos III de Madrid, Spain, 2015. [Online]. Available: https://guti.uc3m.es/data/_uploaded/thesis/PhD_Thesis_BFS.pdf (accessed Mars. 10, 2025).

[51] Synaptics, "Fingerprint sensing biometric security," [PDF]. [Online]. Available : https://www.synaptics.com/sites/default/files/fingerprint-sensing-biometric-security.pdf (accessed Mars. 12, 2025).

[52] Freemindtronic, "Fingerprint security risks – DataShielder HSM,". [Online]. Available: https://freemindtronic.com/fingerprint-security-risks-datashielder-hsm/ (Accessed Mars. 14, 2025).

[53] PowerTech DZ, "Carte de développement ESP32 - 38 broches,". [Online]. Available: https://powertech-dz.net/products/single/carte-de-developpement-esp32-esp-32-38pin-vente-composants-electronique-blida-algerie-50 (accessed Mars. 16, 2025).

[54] Robot.com.ve, "ESP-WROOM-32 module WiFi & Bluetooth,". [Online]. Available: https://robot.com.ve/product/tarjeta-de-desarrollo-esp-wroom-32-esp322-wifi-bluetooth/ (accessed Mars. 19, 2025).

[55] Arduino Yard, “ESP32 Pinout and GPIO Reference Guide,” *ArduinoYard*. [Online]. Available: https://arduinoyard.com/esp32-pinout-guide/ (accessed Jun. 8, 2025).

[56] Industrial Shields, "ESP32 Bluetooth BLE & WiFi," Apr. 10, 2025. [Online]. Available : https://www.industrialshields.com/blog/arduino-industrial-1/esp32-bluetooth-ble-wifi-133 (accessed Mars. 02, 2025).

[57] IRJMETS, "Fingerprint-based biometric security system," [PDF]. [Online]. Available: https://www.irjmets.com/uploadedfiles/paper/issue_3_march_2024/50473/final/fin_irjmets1710484789.pdf (accessed Mars. 20, 2025).

[58] Espressif Systems, "ESP32 datasheet," [PDF]. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (accessed Mars. 22, 2025).

[59] Brahimi and H. Guezouli, "Étude et réalisation d’une carte de commande à base d’un microcontrôleur PIC 16f877 pour ponts redresseurs triphasés à thyristors," Master’s thesis, Univ. Abou-Bekr Belkaïd Tlemcen, 2013–2014. (Accessed.Apr. 11, 2025).

[60] ElectronicWings, "LCD 16x2 interfacing with Arduino Uno,". [Online]. Available: https://www.electronicwings.com/arduino/lcd-16x2-interfacing-with-arduino-uno (accessed Mars. 25, 2025).

[61] Robotique.tech, "Définition de LED,". [Online]. Available: https://www.robotique.tech/tutoriel/definition-de-led/(accessed Mars. 28, 2025).

[62] Instructables, "Blinking an LED with ESP32," [Online]. Available: https://www.instructables.com/Blinking-an-LED-With-ESP32/ (accessed April. 02, 2025).

[63] Robotique.tech, "Tutoriel buzzer," [Online]. Available: https://www.robotique.tech/tutoriel/buzzer/ (accessed April. 23, 2025).

[64] Pija Education, "4x3 keypad operating mechanism,". [Online]. Available: https://pijaeducation.com/arduino/keypad/4x3-keypad-operating-mechanism/ (accessed April. 05, 2025).

[65] ElectroPeak, "Interfacing 4x3 membrane matrix keypad with Arduino,". [Online]. Available: https://electropeak.com/learn/interfacing-4x3-membrane-matrix-keypad-with-arduino/ (accessed April. 06, 2025).

[66] DIYI0T, "Best battery for ESP32 NodeMCU,". [Online]. Available: https://diyi0t.com/best-battery-for-esp32-nodemcu/ (accessed April. 10, 2025).

[67] Himax Electronics, "32700 LiFePO4 3.2V 6000mAh,". [Online]. Available: https://himaxelectronics.com/product-item/32700-lifepo4-3-2v-6000mah/ (accessed April. 13, 2025).

[68] Random Nerd Tutorials, "ESP32 UART communication (Serial) with Arduino IDE,". [Online]. Available: https://randomnerdtutorials.com/esp32-uart-communication-serial-arduino/ (accessed April. 15, 2025).

[69] TechRM, "Practical guide to ESP32 communication protocols,". [Online]. Available: https://www.techrm.com/practical-guide-to-esp32-communication-protocols/ (accessed April. 18, 2025).

[70] ElectronicWings, "ESP32 Wi-Fi basics - Getting started,". [Online]. Available: https://www.electronicwings.com/esp32/esp32-wi-fi-basics-getting-started (accessed April. 21, 2025).

[71] ElectronicWings, "Servo motor,". [Online]. Available: https://www.electronicwings.com/sensors-modules/servo-motor (accessed April. 22, 2025).

[72] W3Schools, "C++ Introduction,". [Online]. Available: https://www.w3schools.com/cpp/cpp_intro.asp (accessed April. 25, 2025).

[73] Arduino Blaise Pascal, "Logiciel Arduino,". [Online]. Available: https://arduino.blaisepascal.fr/logiciel/ (accessed April. 28, 2025).

[74] Univ. d'Annaba, "Mémoire d'ingéniorat en informatique," [PDF]. [Online]. Available: https://biblio.univ-annaba.dz/ingeniorat/wp-content/uploads/2022/04/memoire.pdf (accessed May. 02, 2025).

[75] Soldered, "Arduino IDE variables,". [Online]. Available: https://soldered.com/learn/arduino-ide-variables/ (accessed May. 03, 2025).

[76] Adafruit, "Adafruit Fingerprint Sensor Library,". [Online]. Available: https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library (accessed May. 04, 2025).

[77] Arduino, "WiFi library,". [Online]. Available: https://docs.arduino.cc/libraries/wifi/ (accessed May. 06, 2025).

[78] Arduino, "Firebase ESP32 Client library,". [Online]. Available: https://docs.arduino.cc/libraries/firebase-esp32-client/ (accessed May. 07, 2025).

[79] Arduino, "HTTPClient library,". [Online]. Available: https://docs.arduino.cc/libraries/httpclient/ (accessed May. 09, 2025).

[80] TechTarget, "RESTful API - Definition,". [Online]. Available: https://www.techtarget.com/searchapparchitecture/definition/RESTful-API (accessed May. 11, 2025).

[81] Arduino Stack Exchange, "ESP32 documentation for time.h,". [Online]. Available: https://arduino.stackexchange.com/questions/71797/esp32-documentaiont-for-time-h (accessed May. 16, 2025).

[82] Elektronique.fr, "Proteus (ISIS et ARES) - Logiciel électronique,". [Online]. Available: https://www.elektronique.fr/proteus-isis-ares (accessed May. 18, 2025).

[83] GeeksforGeeks, "Firebase – Introduction,". [Online]. Available: https://www.geeksforgeeks.org/firebase-introduction/ (accessed May. 21, 2025).

[84] Codes, "Build a website with Google Firebase from scratch," Medium, Sep. 6, 2021. [Online].Available: https://adityacodes.medium.com/build-a-website-with-google-firebase-from-scratch-dd95a99c8d38 (accessed May. 24, 2025).

[85] Losari, "Firebase Realtime Database with many-to-many relationship schema," Mar. 22, 2021.[Online].Available: https://www.alfianlosari.com/posts/firebase-realtime-database-with-many-to-many-relationship-schema/ (accessed May. 25, 2025).

[86] Google,"Realtime Database documentation,". [Online]. Available: https://firebase.google.com/docs/database?hl=fr (accessed May. 28, 2025).

[87] GeeksforGeeks, "Design a web page using HTML and CSS,". [Online]. Available: https://www.geeksforgeeks.org/design-a-web-page-using-html-and-css/ (accessed May. 28, 2025).

[88] GeeksforGeeks, "What is HTTP?". [Online]. Available: https://www.geeksforgeeks.org/what-is-http/ (accessed May. 29, 2025).