BINDURA UNIVERSITY OF SCIENCE EDUCATION

FACULTY OF SCIENCE AND ENGINEERING

DEPARTMENT OF PHYSICS AND ENGINEERING

ELECTRONIC ENGINEERING UNIT

MICROPROCESSOR AND IOT BASED BIOMETRIC DOOR LOCKING SYSTEM

BY

NGWENYA MELUSI

B1851917

SUPERVISOR: Ms E. MUKACHANA

A RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE HONOURS DEGREE IN ELECTRONIC ENGINEERING.

JUNE 2023

APPROVAL FORM

The undersigned certify that they have supervised the student Melusi Ngwenya's dissertation entitled, "Microprocessor and IoT Based Biometric Access Control" submitted in partial fulfilment of the requirements for a Bachelor of Science Honors Degree in Electronic Engineering at the Bindura University of Science Education.

STUDENT:……………………. **DATE:…………………………..**

SUPERVISOR: ……………………… **DATE:…………………….**

CHAIRPERSON: ………………….. **DATE:………………………**

EXTERNAL EXAMINER:..………………… **DATE:…………………….**

ABSTRACT

Biometric authentication has become a popular method of access control due to its high level of security and convenience. Raspberry Pi, a low-cost, credit-card sized computer, has gained popularity in recent years due to its versatility and accessibility. Our proposed system utilizes a Raspberry Pi to capture and process biometric data, such as fingerprints or facial recognition, and grant access to authorized individuals. This system offers a cost-effective and efficient solution for access control in various settings, including offices, schools, and residential buildings. We will further discuss the technical details and implementation of this system in our full paper.

Dedication

For my mother, Miss E Bhondai. You are appreciated.

ACKNOWLEDGEMENTS

First and foremost, I would like to give thanks to the Almighty God in heaven for granting me this special favour of being part of an institution like Bindura University of Science Education. Secondly, I would like to thank my supervisor Mrs Mukachana and the Faculty of Science and Engineering for the unwavering support and guidance that enabled the smooth flow of this project. I would also like to thank my family for their moral and emotional support that they give me. Last but not least, I would like to appreciate my fellow engineers, BUSE Electronic

Engineering Group, special mention goes to Tendai Munetsi, Tafara Mateyo, Brighton Nyadongo and Tarisai Tarupiwa for their great motivation and invaluable input during the course of our programme.

Table of Contents

Table of figures

CHAPTER 1: INTRODUCTION

1.1 : Background

The issue of security of property and human life is an important aspect for individuals, organisations and nations at large. The most common way of securing premises and private property is through a robust access control system. Access control systems provide security by preventing unauthorized access by unregistered individuals and only allowing limited access to registered users according to their prerogative in the system [1]. Access control is generally achieved through locks on entrances to premises which can be doors or gates. Over the years there has been a number of changes to the way security is applied to a system. However, with the advances in industrialisation and urbanisation some of the methods used have become obsolete as they can be compromised with minimum effort [3]. The conventional method of lock and key comes with many disadvantages that include risk of the key being lost, stolen or broken in which case access to the premises is made virtually impossible. This in turn has led to the need to discover more advanced ways of controlling access on different levels. Besides gaining access to a building, lock system can also be used to provide or prevent access to personal belongings and documents both at home or in the office. Some of these belongings could be confidential documents, money, expensive jewelleries which can be kept in a secure vault, safe or strong room. To militate against these challenges, some security systems have been developed to prevent unauthorized access, such as, the use of smartcards, passcode, voice recognition, Radio Frequency Identification (RFID) and biometrics [4]. The main focus of the work done was on the use of Biometric locking system. Biometrics pertains to the science of statistically analysing human biological features like voice, facial recognition fingerprints and other unique characteristics of an individual [5]. The conditions for choosing the criterion are, but not limited to, uniqueness, permanence, collectability, acceptability, performance and or bypassing. Different characteristics such as fingerprint, facial features, eye features and or voice can be used by biometrics. A fingerprint and facial recognition system is looked into in this work. A safe and reliable security system is one of the primary concerns for any organisation. The conventional method of lock and key comes with many disadvantages that include risk of the key being lost, stolen or broken in which case access to the premises is made virtually impossible. This makes a biometric door lock the most suitable application since it eliminates most of the disadvantages of the conventional door lock.

Our goal was to create a system that could accurately identify individuals based on their facial features. To achieve this, we utilized a machine learning algorithm called OpenCV which is well-suited for image recognition tasks. Training model on a dataset of facial images, was completed using a technique called transfer learning to improve the accuracy of our system. The Python code was also optimized for the Raspberry Pi platform to ensure efficient processing of the images. The experimental results demonstrate that the system achieved high accuracy in identifying individuals from a set of facial images. The project can have practical applications in security systems, access control, and other areas where facial recognition is required. The biometric recognition on the Raspberry Pi platform is a promising step towards developing a reliable and efficient system for identifying individuals based on their facial and fingerprint features.

1.2 : Problem Statement

The current door locking system at the institution makes use of RFID tags to open locked doors. The doors are locked using an electromagnetic lock which automatically unlocks when the tag is brought in close proximity to the tag reader and a hydraulic arm is used to return the door with the armature plate back to the electromagnet thereby locking the door.

1.3 : Justification

This therefore was the main challenge that the design project sought to address by using the biometric locking system.

1.4 : Aim

- To design a biometric access control system which makes use of face and fingerprint features to grant access
- To ensure unauthorized users are denied access to the premises

1.5 : Objectives

- ✓ Ensure safety and security of the organisation’s premises.
- ✓ Provide a more reliable security system.
- ✓ To ensure fingerprint authentication on the system
- ✓ To ensure facial recognition on the system
- ✓ IoT support for the system to record and monitor access
- ✓ Write and compile code for the programmes

1.6 : Assumptions

- Users are using Windows 10 as their operating system.
- Users can read and communicate in English.
- Users are using the system to grade and classify tobacco images.
- Users are using Windows as their operating system preferably Windows 10.
- Users can read and communicate in English.
- Users are using the system for access control.

1.7 : Limitations

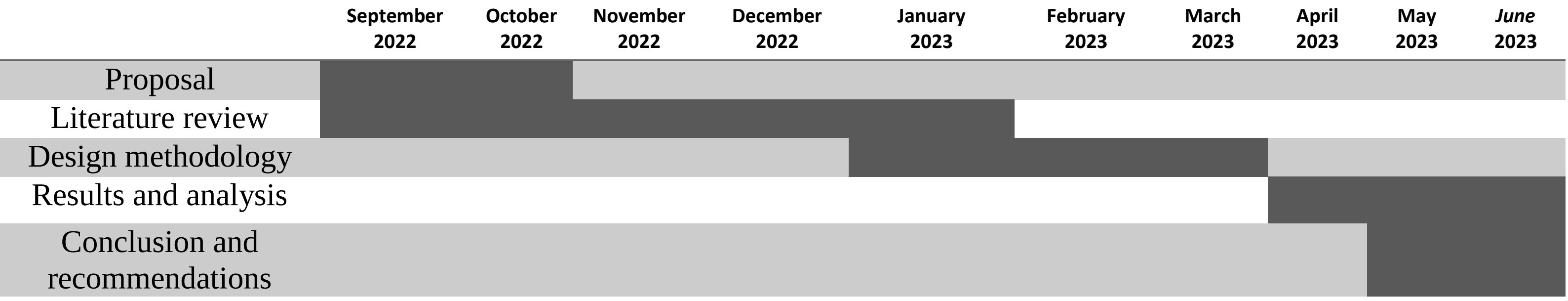
Some of the restrictions that the engineer came across during the course of the project design include the following:

- ➢ **Time:** The limited time in which the research is to be conducted.
- ➢ **Unavailable Libraries:** The project is programmed using Python and also Proteus Simulation Software. Some libraries that are required cannot be supported by these tools since they are third-party-owned. Thus, some packages might bring compilation challenges.
- ➢ **Limited Resources:** The project would have produced impeccable results with the use of some specific hardware and software tools that demand more computational power. In light of the unforeseen economic challenges, the researcher opted to use lightweight algorithms for the purpose of developing the prototype.

- **Network Access:** The system is developed primarily for Raspberry Pi running the 32bit version of the Legacy Operating system. It operates as a stand-alone server for IoT capabilities which requires some form of network access to operate.

1.8 Project Gantt Chart

A request to see the timeline of the development of the system was made by the manger so a Gantt chart was constructed showing a detailed timeline of production of the project.



CHAPTER 2: LITERATURE REVIEW

2.1 : Introduction

The main aim of this chapter was to explain the theory behind the biometric door lock's design. This included the general overview of alternative door locking systems. Functional block diagrams of the different door locking systems were used to analyse their physical setup and standard operating procedure.

2.2 : Alternative Solutions

There are various door locking system designs that are used in the making of access control systems. The type of locking mechanism used depends on a number of factors that include application of the door lock and the availability of resources. The four main access control designs are discussed in further detail below.

i. Lock and Key System

The conventional method of access control for many entrances is the lock and key system. his system is widely used in homes, offices, and other buildings for security purposes [13]. The lock and key mechanism consist of a lock cylinder and a key that fits into the cylinder. When the key is turned, the lock cylinder rotates, allowing the door to be opened or closed. There are different types of lock and key door locks, including pin-tumbler locks, wafer-tumbler locks, and disc-tumbler locks. Each type has its own unique features and advantages. For example, pin-tumbler locks are the most common type of lock and key system and are known for their reliability and durability. One of the main advantages of the lock and key system is its simplicity. It is easy to understand and use, and does not require any special skills or knowledge. However, this simplicity also makes it vulnerable to picking and other forms of attack. To address these vulnerabilities, manufacturers have developed more advanced lock and key systems, such as electronic locks and biometric locks. These systems use advanced technology to provide greater security and convenience. Overall, the lock and key door lock system remains an important and widely used security measure in buildings of all types. Its simplicity, reliability, and durability make it a popular choice for homeowners and businesses alike. The responsible individual carries their own copy of the key which they can use to unlock the door. For banking institutions this conventional method makes use of a dual controlled two-key lock that requires two keys to unlock at one time for the door to open. This method is ideal in that we can have a key for the Branch manager and another for the Operations manager. This will ensure that for the door to be opened both individuals may be required to be present. This improves the first level of security at the entrance of the bank.

Figure 2.1: Lock and Key

The lock-and-key system is supplemented with an additional locking mechanism like a keypad lock or an RFID lock. This is done to ensure the door remains locked using a heavy-duty magnetic door lock in case the keys have been accessed by unauthorised personnel.

ii. Keypad Door Lock

This method of access control makes use of a Personal Identification Number (PIN) to unlock the door. Each user may have their own PIN which they can use to unlock the door and gain access to the building. Keypad door locks have become increasingly popular due to their convenience and security features [8]. We analyze the different types of keypad door locks available in the market, including mechanical and electronic locks. We also discuss the advantages and disadvantages of using keypad door locks in different environments, such as residential and commercial buildings. Furthermore, we review the existing research on the effectiveness of keypad door locks in preventing unauthorized access. Our findings suggest that keypad door locks are not that reliable and efficient a means of securing entryways because of the possibility of PIN sharing.

Figure 2.2: Keypad Lock

A keypad matrix is attached to a microcontroller unit which inputs, stores and processes the PIN entered by the user. The microcontroller unit then gives an output signal to either lock or unlock the door based on the validity of the PIN entered.

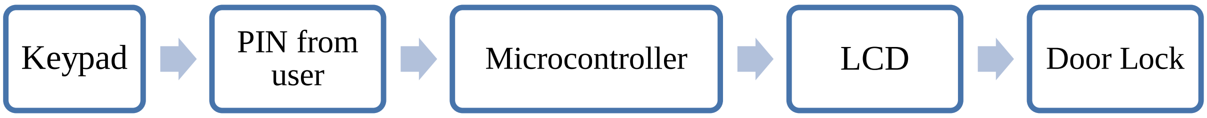


Figure 2.3: Keypad Locking System

iii. RFID Door Lock

Radio Frequency Identification (RFID) is also another method used for applications like access control. This type of access control device provides a consistent and reliable input, storage and processing of trackable data. RFID locking systems are contactless, meaning that the credential doesn't have to touch the reader for it to work. The readers work by sending and receiving data which is transmitted over radio frequencies [13]. An RFID door locking system requires RFID tags, antennas, an RFID reader, and a transceiver in order to function as a complete system. The use of RFID in door locks has gained popularity in recent years due to its convenience and security benefits. RFID door locks work by using a small electronic device, called a tag or a card, which contains a unique identifier. When the tag is placed near the door lock, the lock reads the identifier and grants or denies access based on the permissions associated with that identifier. One of the main advantages of RFID door locks is their ease of use. Unlike traditional locks that require physical keys, RFID locks can be operated with a simple wave of a card or tag. This makes them ideal for environments where a large number of people need access, such as office buildings or hotels. Another advantage of RFID door locks is their security. Since each tag or card contains a unique identifier, it is much more difficult for unauthorized individuals to gain access. Additionally, access permissions can be easily managed and updated, further enhancing security.

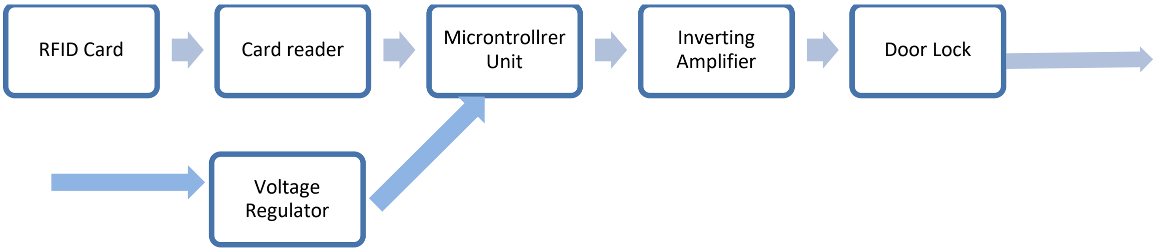


Figure 2.4 – RFID Door Lock System

Figure 2.5 - RFID Reader and tags

2.3 : Chosen Solution

Based on the comparison and evaluation of the solutions given above the company found the Microprocessor Based Biometric Door Locking System as the most favourable solution.

The Biometric Door Locking System was chosen because it offers the following benefits over the other alternatives:

- ✓ It is relatively more secure since biometrics cannot be duplicated or stolen
- ✓ It focuses mainly on solving the current problems being faced at the organisation
- ✓ It provides a log of who accessed the system either successfully or unsuccessfully
- ✓ Errors in the system can be solved easily
- ✓ User requirements can be satisfied

CHAPTER 3: DESIGN

3.1 : Introduction

In this chapter, the main design specification of the biometric access control system was done. The main system design was centred on the Raspberry Pi 4 Model B which uses the Raspbian Operating system. The Raspberry Pi was first configured to access the peripherals that were connected to the Serial communication port and the GPIO pins.

3.2 Requirements Analysis

In House Development of the system was the most favourable because it has the following advantages: -

- ✓ It is cheaper
- ✓ It focuses mainly on solving the current problems
- ✓ Errors in the system can be solved fast and easily
- ✓ User requirements can be satisfied

Hardware Requirements

The chosen solution has the following hardware requirements:-

HARDWARE REQUIRED	COMMENT
21" MONITOR	DISPLAYS CLEAR OUTPUT EVEN IF THE USER IS SHORT SIGHTED
QWERTY KEYBOARD	FOR ENTERING IN TEXT. IT IS EASY TO USE FASTER THAN WRITING WITH A PEN AND CORRECTIONS ARE MADE EASILY
MOUSE	FOR NAVIGATION. EASY ACCESS TO DOCUMENTS AS IT MOVES AROUND
ETHERNET CABLES/TWISTED PAIR CABLES	FOR CONNECTING RASPBERRY PI TO NETWORK HUB

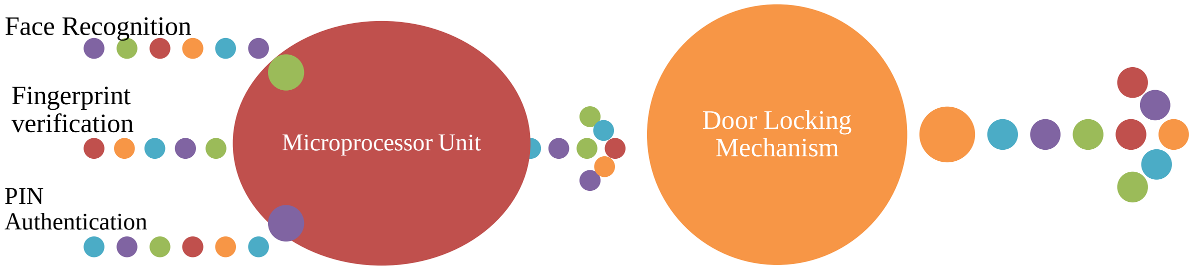
Software Requirements

The chosen solution has the following hardware requirements: -

SOFTWARE REQUIRED	COMMENT
Raspberry Pi Os Debian Buster (Legacy)	Suitable for running face recognition and fingerprint verification
Thonny IDLE	For coding the python programs
Putty	Terminal Emulator used as an SSH Client
VNC Server/VNC Viewer	For Connecting to Raspberry Pi Server via the local network

3.2.1 Operation of the Biometric Door Locking Systems

This report gives an insight into the operational requirements of the Biometric door locking system. The system includes a smart and affordable door lock enhanced with a face, fingerprint and pin interface. The main components used in implementing this project include the Raspberry Pi 4 Model B, a camera module, an JM101 fingerprint sensor, a 4x4 keypad matrix, the door locking mechanism. The biometric sensors scan the feature, either face or fingerprint and then compare it to the stored data in the database of our microprocessor unit. If there is a match with one of the faces or fingerprints in memory, the Raspberry Pi will transmit an instruction to unlock the electromagnetic door lock. If no match is found a message is displayed on the LCD informing the user to retry scanning of face or fingerprint.



1. Raspberry Pi 4 Model B Microprocessor Unit

The Raspberry Pi Model 4 B is the latest iteration of the popular single-board microprocessor computer box [16]. The Model 4 B boasts significant improvements over its predecessor, including a faster processor, increased memory capacity, and enhanced connectivity options.The model 4 b delivers a marked improvement in performance, making it ideal for a wide range of applications, from home automation to industrial control systems. Additionally, the model 4 b features up to 8 GB of LPDDR4-3200 SDRAM, allowing for increased multitasking capabilities and improved performance in memory-intensive applications. The model 4 b also includes Gigabit Ethernet, dual-band 802.11ac wireless, Bluetooth 5.0, and two USB 3.0 ports, providing users with a variety of connectivity options. Overall, the Raspberry Pi model 4 b represents a significant improvement over its predecessor and is a versatile and powerful tool for a wide range of applications. he Model 4 B features a Broadcom BCM2711 SoC with a quad-core Cortex-A72 CPU running at 1.5GHz, up to 8GB of LPDDR4-3200 SDRAM, dual-band 802.11ac wireless networking, Bluetooth 5.0, Gigabit Ethernet, two USB 3.0 ports, two USB 2.0 ports, and two micro-HDMI ports supporting up to 4Kp60 resolution. Additionally, the Model 4 B includes a 40-pin GPIO header and a microSD card slot for storage. Overall, the Raspberry Pi Model 4 B is a powerful and flexible single-board computer that is well-suited for a wide range of applications.

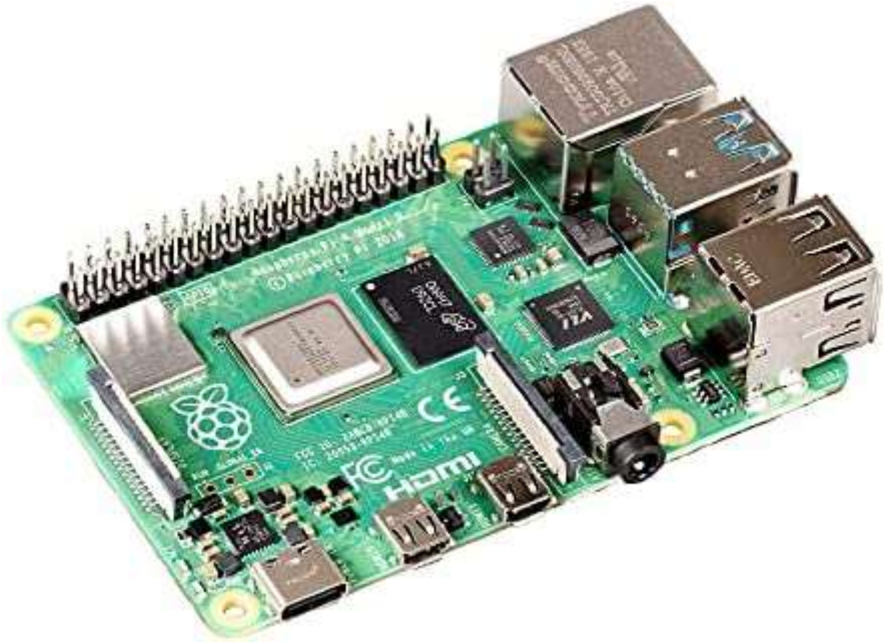


Figure 3.1 – Raspberry Pi Model 4B

2. Raspberry Pi Camera Module

This camera module is a small and affordable device that can be easily connected to a Raspberry Pi board [16]. It is capable of capturing high-quality images and videos with a resolution of up to 8 megapixels and 1080p respectively. The module is equipped with a Sony IMX219 sensor that has a 1/4-inch optical format and a pixel size of 1.12 micrometres. It also features a fixed-focus lens with an aperture of f/2.0 and a field of view of 62.2 degrees. The camera module can be controlled using the Raspberry Pi camera software, which provides a user-friendly interface for capturing and processing images and videos. Additionally, the module supports various modes of operation, including still image capture, video recording, and time-lapse photography. We discuss the technical specifications of the camera module, its performance characteristics, and its potential applications in various fields, such as robotics, surveillance, and scientific research. Overall, the Raspberry Pi camera module is a versatile and cost-effective solution for capturing visual data in a wide range of applications.

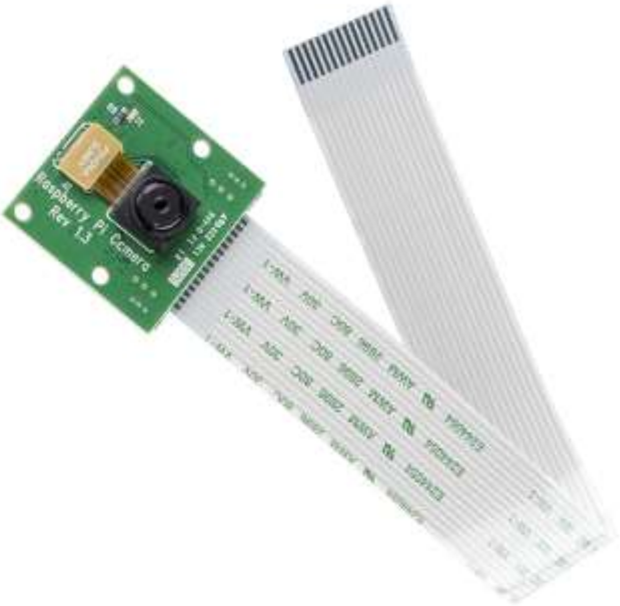


Figure 3.2– Raspberry Pi Camera Module

Provides high sensitivity, low crosstalk and low noise image capture in an ultrasmall and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the processor.

3. JM101 Fingerprint Sensor Module

This module is a compact and efficient device that allows for the easy integration of fingerprint recognition technology into various applications [17]. The JM101 module utilizes a high-resolution optical sensor that captures high-quality fingerprint images. These images are then processed by the module's advanced algorithms to extract unique fingerprint features, which are then used for identification or verification purposes. One of the key advantages of the JM101 module is its small size and low power consumption. This makes it an ideal solution for portable devices such as smartphones, tablets, and laptops. Additionally, the module's robust design ensures reliable performance even in harsh environments. The JM101 module supports a variety of communication interfaces, including UART, USB, and SPI. This allows for easy integration with a wide range of microcontrollers and other devices. The module also includes a software development kit (SDK), which provides developers with a comprehensive set of tools for integrating fingerprint recognition into their applications. Overall, the JM101 fingerprint module is a powerful and versatile solution for adding fingerprint recognition capabilities to a wide range of devices and applications. Its compact size, low power consumption, and robust design make it an ideal choice for developers looking to add biometric security to their products. The module also has the capacity to store at least 300 new fingerprints can be enrolled directly and they can be stored in the onboard FLASH memory.

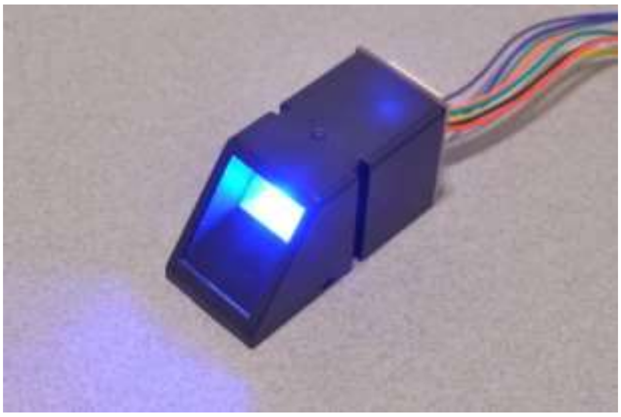


Figure 3.3 : JM101 Fingerprint module

Fingerprint processing includes two parts: fingerprint enrollment and fingerprint matching, the matching can be 1:1 or 1: N. When enrolling, a user needs to enter the finger two times. The system will process the two -time finger images, generate a template of the finger based on processing results, and store the template. When matching, the user enters the finger through an optical sensor, and the system will generate a template of the finger and compare it with templates of the finger library. For 1:1 matching, the system will compare the live finger with a specific template designated in the Module; for 1: N matching, or searching, the system will search the whole finger library for the matching finger. In both circumstances, the system will return the matching result, success or failure.

JM101 Fingerprint Module Connection

3.3V Pin (Red Wire) – Pin 1

GND Pin (Black Wire) – Pin 9

Serial Data Transmit (White Wire) – GPIO14

Serial Data Transmit (Yellow Wire) – GPIO15

4. 16 x 2 Liquid Crystal Display

The 16x2 LCD is a liquid crystal display with 16 columns and 2 rows of characters. It is commonly used in applications that require a simple and cost-effective display solution. The module consists of a controller chip, which communicates with the microcontroller, and a display panel, which consists of 16x2 pixels. The controller chip is responsible for driving the pixels and providing the necessary timing signals. The 16x2 LCD operates on a 5V supply voltage and consumes a low amount of power, making it suitable for battery-powered devices. The module supports a wide range of character sets, including ASCII and Japanese characters. It also has built-in character generator ROM, which allows users to create custom characters. The display panel of the 16x2 LCD is made up of two lines of 16 characters each. Each character is composed of 5x8 pixels, which can display a wide range of characters, symbols, and numbers. The module also includes a backlight, which can be turned on or off depending on the application requirements. The 16x2 LCD is a versatile and cost-effective display module that is widely used in various electronic devices. Its low power consumption, support for multiple character sets, and custom character generation make it a popular choice among designers during the rapid prototype stage of project development.

Figure 3.4 : 16 x 2 Liquid Crystal Display

The 16×2 LCD pin layout is given below:

- Pin 1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin 2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin 3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin 4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin 5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

5. 4x4 Matrix Keypad

The keypad consists of 16 keys arranged in a 4x4 matrix pattern, where each key corresponds to a unique combination of row and column connections. The keypad is commonly used in security systems, ATMs, and other applications that require user input. The electrical connections of the keypad are established using a matrix of conductive traces, which are usually printed on a flexible circuit board. The traces are arranged in a way that each key is connected to a unique combination of row and column traces. When a key is pressed, it connects the corresponding row and column traces, which can be detected by the electronic system. The keypad can be interfaced with a microcontroller using various techniques, such as polling and interrupt-driven methods. In the polling method, the microcontroller periodically scans the rows and columns of the keypad to detect any key presses. In the interrupt-driven method, the microcontroller is interrupted whenever a key is pressed, which reduces the processing time and power consumption. To improve the security of the system, the keypad can be combined with other input devices, such as biometric sensors or smart cards. The keypad can also be programmed to detect and prevent various types of attacks, such as brute-force attacks and replay attacks. In conclusion, the 4x4 matrix keypad is a versatile and reliable input device that can be used in various electronic systems. Its simple design and low cost make it an attractive option for many applications. However, its security can be enhanced by combining it with other input devices and implementing appropriate security measures.

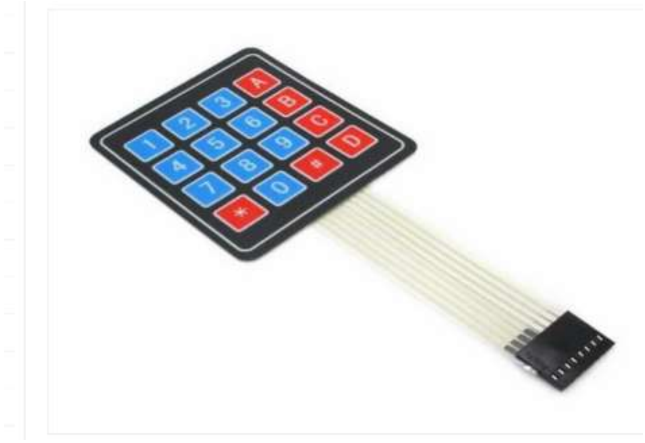


Figure 3.1: 4x4 Matrix Keypad

6. Ultrasonic Range Sensor

The HC-SR04 ultrasonic sensor is a popular choice for distance measurement in robotics and automation applications [15]. It operates by emitting ultrasonic waves and measuring the time it takes for the waves to bounce back off an object. The sensor consists of a transmitter and a receiver, and can measure distances up to 4 meters with an accuracy of 3 mm. The HC-SR04 is easy to use and relatively inexpensive, making it a popular choice for hobbyists and professionals alike. However, it is important to note that the sensor can be affected by environmental factors such as temperature and humidity, which can impact its accuracy. The Echo pin is connected to the Raspberry Pi using a voltage divider circuit which drops its potential from 5V to 3.3V. this is done because the GPIO pin it is connected to requires 3.3V to operate and anything greater than that may cause the system to break.

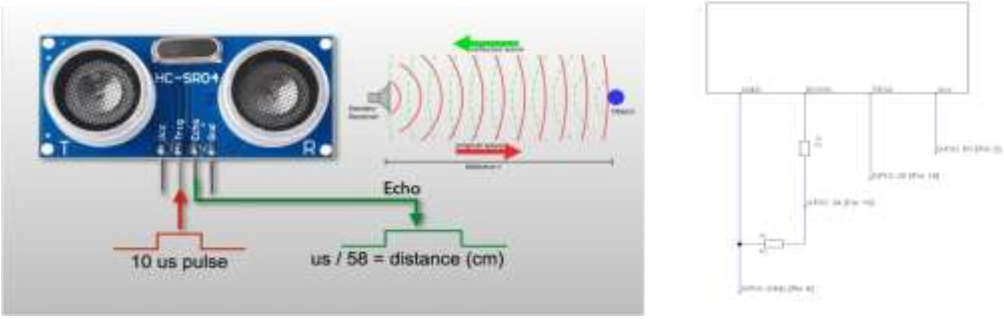


Figure 3.6 – HC-SR04 Range Sensor

Furthermore, we analyzed the limitations of the HC-SR04 sensor. One limitation is its narrow detection angle, which can lead to inaccurate readings if the object is not directly in front of the sensor. Another limitation is its inability to detect transparent or reflective objects, as the ultrasonic waves can pass through or bounce off these objects without being detected.

7. One Channel Relay

The one channel relay, is a device used to control the flow of electricity in a circuit [17]. By using a one channel relay with a magnetic door lock, we can create a more secure and efficient system for controlling access to a building or room. The relay can be used to control the power supply to the magnetic lock, allowing it to be opened or closed as needed. This can be done remotely, using a control circuit from the Raspberry Pi 4. One advantage of using a one channel relay with a magnetic door lock is that it allows for greater flexibility in the control of access.

For example, different users can be given different levels of access, depending on their needs and permissions. This can be done through the use of different relays, each controlling a different aspect of the system. Another advantage of using a one way relay with a magnetic door lock is that it can help to prevent unauthorized access. By controlling the power supply to the lock, the relay can ensure that only authorized users are able to gain entry. This can be particularly important in high-security settings, such as government buildings or research facilities.

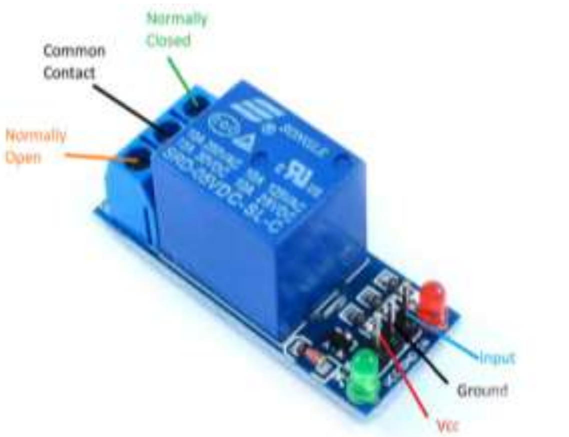


Figure 2.7: One Way Relay

The use of a one-way relay with a magnetic door lock can provide a secure and efficient system for controlling access to a building or room. By using different relays, access can be customized to meet the needs of different users, while also ensuring that only authorized users are able to gain entry.

8. Electromagnetic Door Lock

An electromagnetic door lock is used to keep a trap door normally closed when an electric current is being supplied to it [17]. It makes use of a very strong electromagnet attached to the frame and an armature plate fixed to the door. It's basically an electronic lock, designed for heavy duty use since the electromagnet has a holding force between 30kg and 600kg. this makes it ideal for application in high security areas like banks and strong rooms. The operating voltage is 9-12VDC and when it is applied, the electromagnet turns on and attracts the armature plate towards it locking the door in place. To open the door, you simply cut the power supply to the electromagnet which turns it off releasing the armature. It does not use any power in this state. A signal to cut power supply to the electromagnet can be an output from the biometric access control system which is transmitted when a known biometric feature is detected. It is very easy to install for automatic door lock systems like trap doors in a bank.

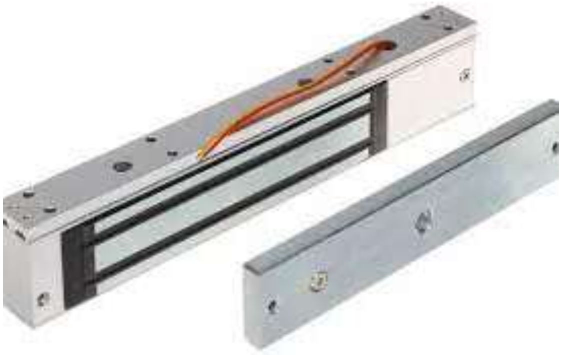


Figure 3.8 - Electromagnetic Lock

Magnetic door locks have become increasingly popular in the banking industry due to their ability to provide a high level of security. These locks use an electromagnetic force to secure doors and prevent unauthorized access. Studies have shown that magnetic door locks are effective in preventing break-ins and robberies. In addition, they are easy to install and maintain. However, it is important to note that magnetic door locks are not foolproof and can be vulnerable to certain types of attacks. To mitigate these vulnerabilities, banks often use additional security measures such as surveillance cameras and alarm systems. It is also important for banks to regularly inspect and maintain their magnetic door locks to ensure they are functioning properly. Overall, magnetic door locks have proven to be a reliable and effective security measure for banks. However, it is important for banks to remain vigilant and implement additional security measures to ensure the safety of their employees and customers.

3.3 : System Flow Chart

To initiation the design stage, the engineer designed flow chart diagrams for the processes which include face recognition, fingerprint verification and PIN authentication. The system flow chart gives an overview of how the system operates and how it reacts under various conditions. Start and End terminals are found at the begin and end respectively of the flow chart. The processes and decisions are found in between the terminals and are connected by arrows.

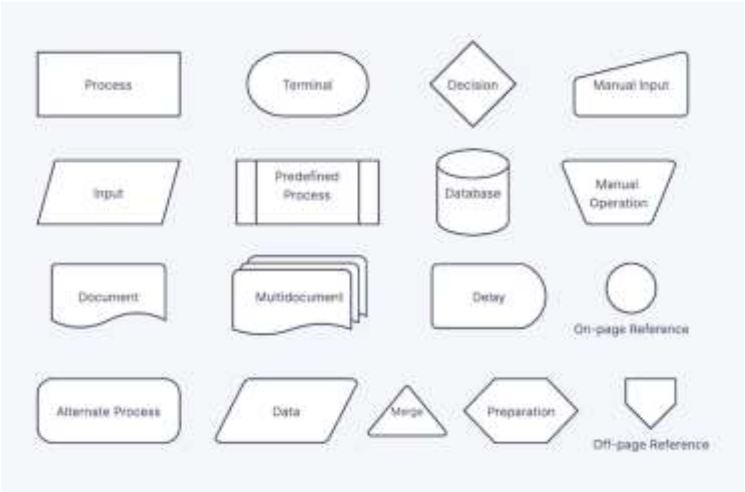
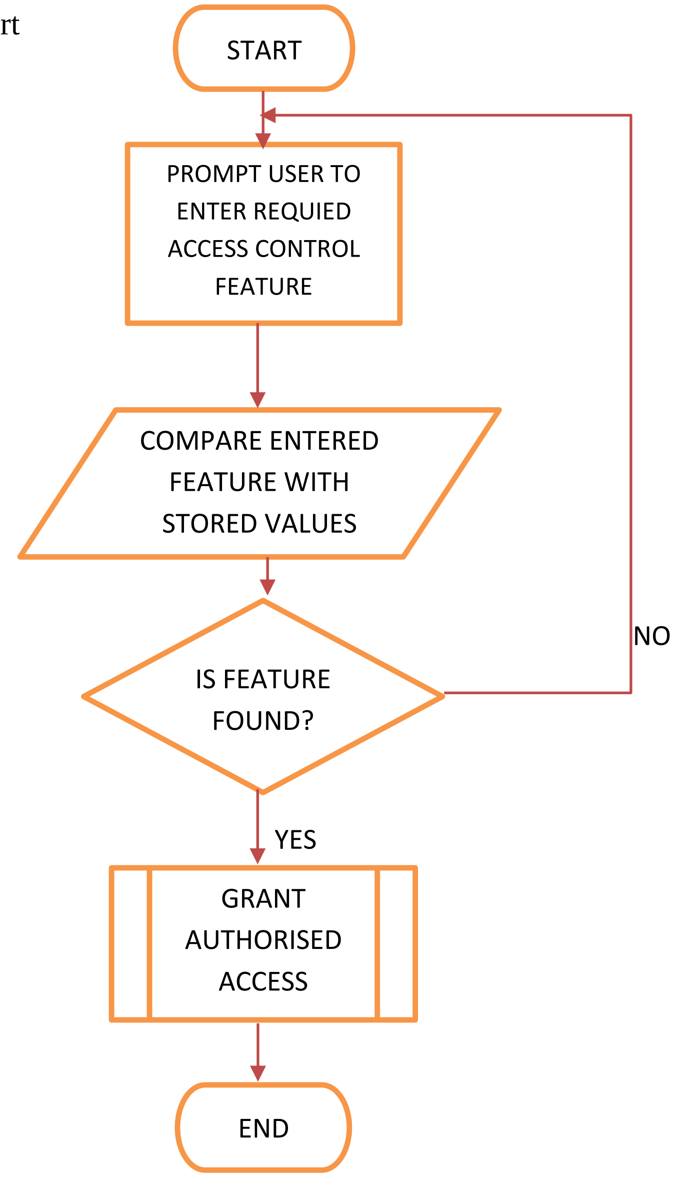


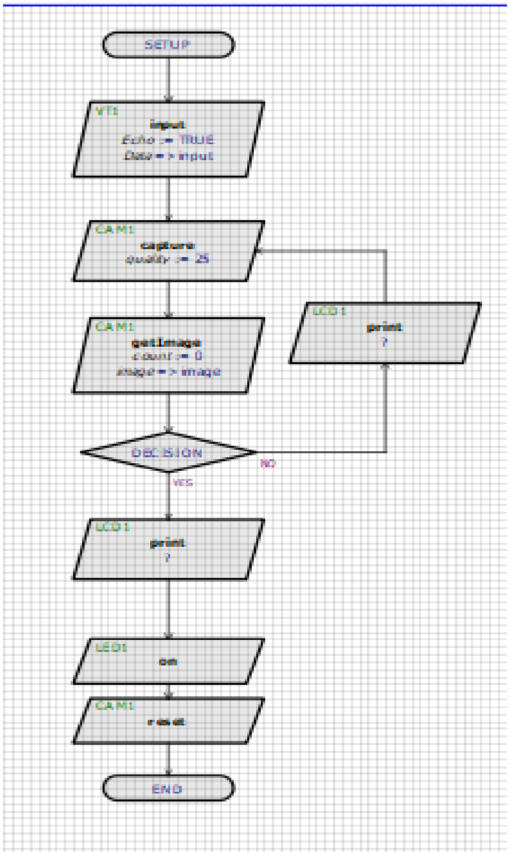
Figure 3.9 – Flow chart symbols

General System Flow Chart


START
PROMPT USER TO ENTER REQUIED ACCESS CONTROL FEATURE
COMPARE ENTERED FEATURE WITH STORED VALUES
IS FEATURE FOUND?
NO
YES
GRANT AUTHORISED ACCESS
END

i. Face Recognition Flow Chart

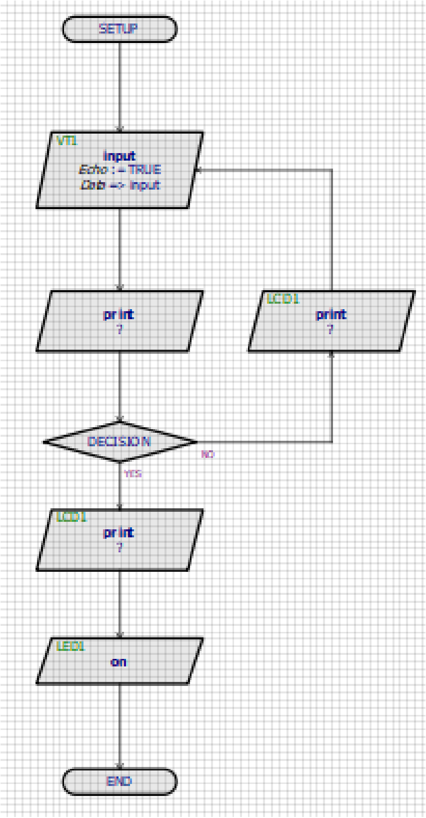
The process of face recognition was represented in a flow chart. The process begins with face detection, where the system locates the face in the image. This is followed by face alignment, which involves normalizing the face orientation and size. Next, we extract facial features which are then used to train a machine learning model, in this case which is OpenCV. During the recognition phase, the input face is compared with the trained model to identify the person. The accuracy of the system can be improved by using multiple images of the same person during training. An LED was used to represent a successful login and door opening.



Once the features have been extracted by the machine learning software, they are compared to a database of known faces to determine the identity of the person in the image. The final step in the process is to output the result of the recognition, which may include the person's name or a confidence score indicating the likelihood of a match which is used to open the door. Overall, the face recognition flow chart provides a clear and concise overview of the steps involved in this complex process. By understanding each step, in detail, the engineer was able to develop more accurate and efficient face recognition system.

ii. Fingerprint Verification Flow Chart

The flow chart consists of a series of steps that are followed to identify an individual based on their unique fingerprint pattern. The first step in the process is to capture the fingerprint image using a sensor. This image is then pre-processed to enhance its quality and remove any noise. The next step involves feature extraction, where the unique features of the fingerprint are identified and extracted. These features are then compared to a database of known fingerprints to identify the individual. The final step is to make a decision based on the comparison results. The flow chart provides a systematic approach to fingerprint recognition, which is essential for its successful implementation in various applications such as access control, identification, and security systems.



As seen in the Figure above, the system utilizes biometric identification techniques to grant access to authorized personnel. The circuit consists of a microcontroller, a fingerprint sensor, a keypad, and a relay module. The microcontroller is responsible for processing the input from the fingerprint sensor and the keypad, and controlling the relay module to grant or deny access. The fingerprint sensor captures the fingerprint of the user and compares it with the stored fingerprint in the system's database. If the fingerprint matches, the user is granted access. If not, the user is denied access. The keypad provides an additional layer of security by requiring a PIN code to be entered along with the fingerprint. The relay module is responsible for controlling the door lock mechanism. Overall, the biometric access control system circuit design presented in this paper offers a secure and reliable method of access control.

3.4 : Proteus Circuit Design

Last but not least was the circuit design using Proteus 8 simulation software. The circuit designed uses fingerprint recognition as the biometric authentication method. The Proteus software was used to simulate and test the circuit design. The circuit consists of a camera module, fingerprint sensor module, a microprocessor unit, and a relay module. The camera and fingerprint sensor module captures the fingerprint image and sends it to the microprocessor unit for processing. The microcontroller unit compares the captured fingerprint with the stored fingerprint in its memory and sends a signal to the relay module to unlock the door if the fingerprints match. The circuit was tested for accuracy and security, and the results were satisfactory. Our design can be used in various applications, such as access control systems for offices, homes, and other secure areas. In conclusion, the biometric access control circuit designed using Proteus is an effective and reliable solution for access control systems.

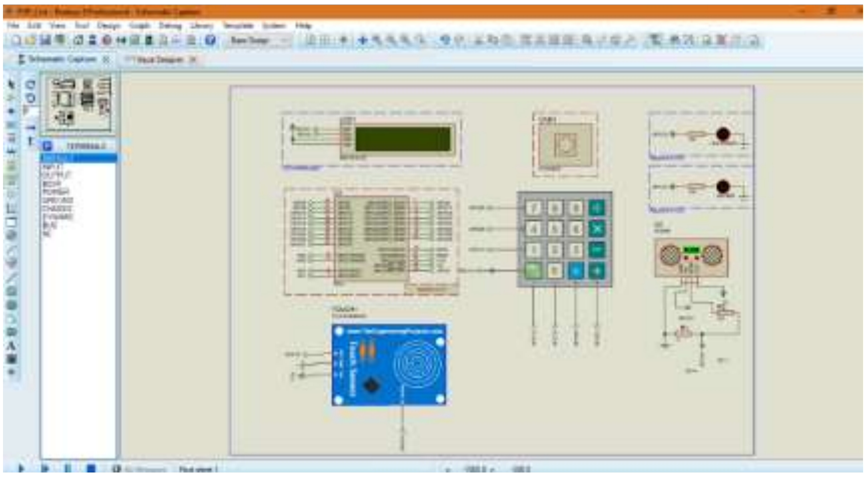
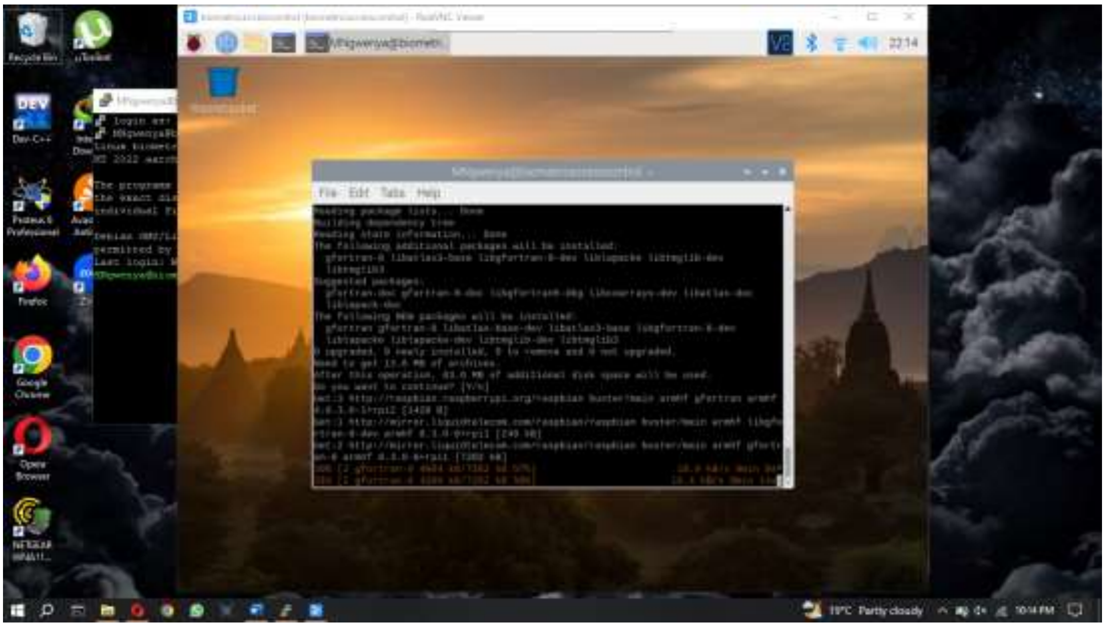


Figure 3.9 – Circuit Design

3.5 : Raspberry Pi Configuration

After the system flow charts were designed, the engineer then went on to configure the Raspberry Pi Model 4B for the different programmes running on it. First and foremost was the installation of the 32bit Raspberry Pi Legacy operating system. This operating system was the most ideal as it provides access to tools like OpenCV which run face recognition machine learning. The Raspberry Pi was also programmed to operate as a stand-alone IOT server running the different programmes. To access the Raspberry Pi remotely you activate Secure Shell (SSH) and provide the login credentials which will be used to access the server. The server can be accessed by any machine connected to the same network and running an SSH client, for example Putty, and a server client software, for example VNC Server and VNC Viewer. After the operating system was installed, the Engineer then began configuration of the OpenCV machine learning system and installation of the relevant libraries.

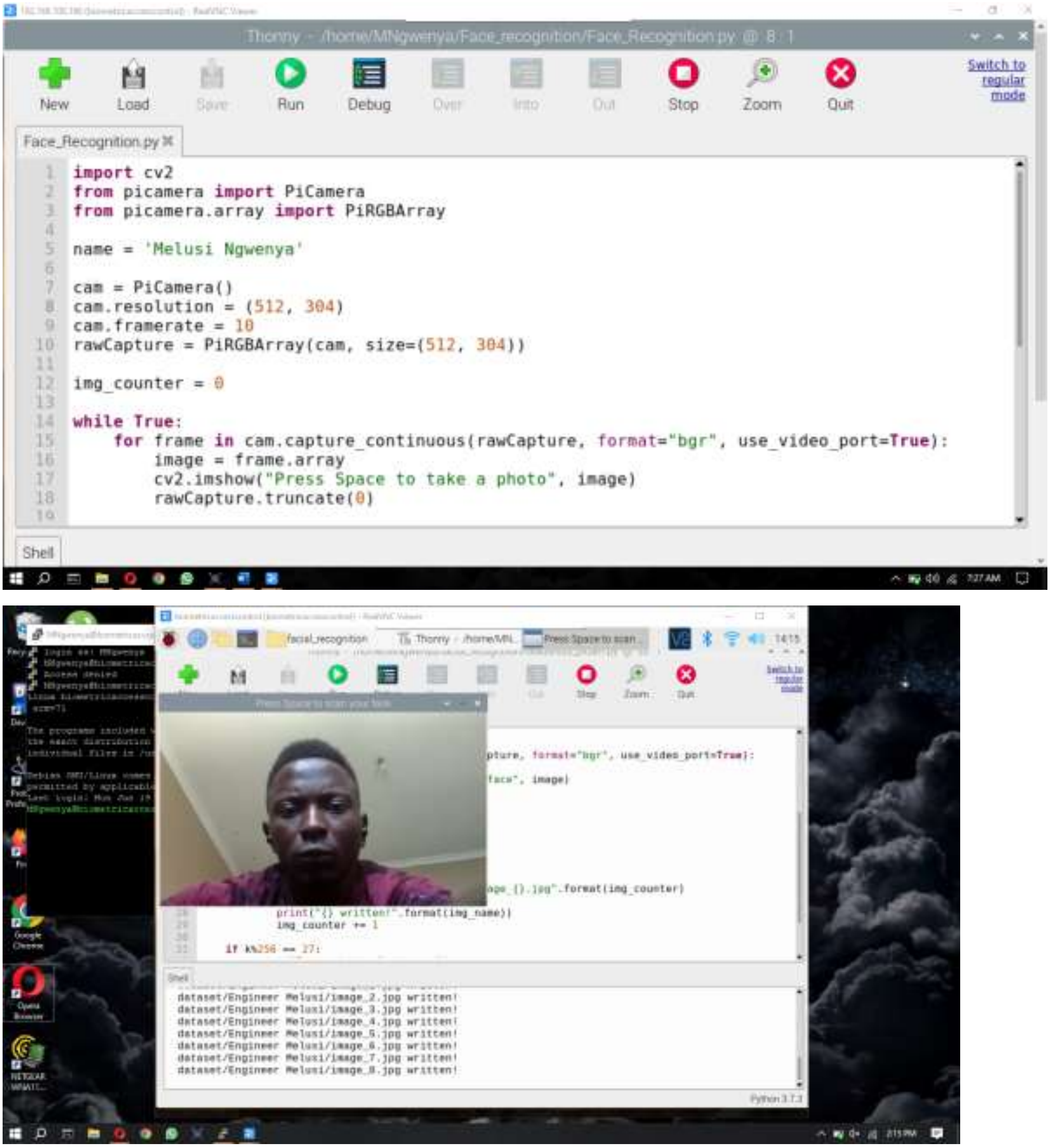


```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gfortran-8 libatlas3-base libgfortran-8-dev liblapacke libtmglib-dev
  libtmglib3
Suggested packages:
  gfortran-doc gfortran-8-doc libgfortran8-dbg libcoarrays-dev libatlas-doc
  liblapack-doc
The following NEW packages will be installed:
  gfortran gfortran-8 libatlas-base-dev libatlas3-base libgfortran-8-dev
  liblapacke liblapacke-dev libtmglib-dev libtmglib3
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 13.6 MB of archives.
After this operation, 83.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://raspbian.raspberrypi.org/raspbian buster/main armhf gfortran armhf
4:8.3.0-1+rpi2 [1428 B]
Get:3 http://mirror.liquidtelecom.com/raspbian/raspbian buster/main armhf libgfo
rtran-8-dev armhf 8.3.0-6+rpi1 [249 kB]
Get:2 http://mirror.liquidtelecom.com/raspbian/raspbian buster/main armhf gfortr
an-8 armhf 8.3.0-6+rpi1 [7202 kB]
```

Raspberry Pi Configuration

i. Face Recognition Machine Learning

To begin the process of training the system you first take pictures of the face required and add them to the face recognition dataset folder. This folder will be used as a local database of the face recognition system where the known faces will be stored. To improve on the training and identification of an individual it recommended to take at least ten pictures of the face at different angles and store them in the user's dataset.



```python
import cv2
from picamera import PiCamera
from picamera.array import PiRGBArray

name = 'Melusi Ngwenya'

cam = PiCamera()
cam.resolution = (512, 304)
cam.framerate = 10
rawCapture = PiRGBArray(cam, size=(512, 304))

img_counter = 0

while True:
    for frame in cam.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        image = frame.array
        cv2.imshow("Press Space to take a photo", image)
        rawCapture.truncate(0)
```

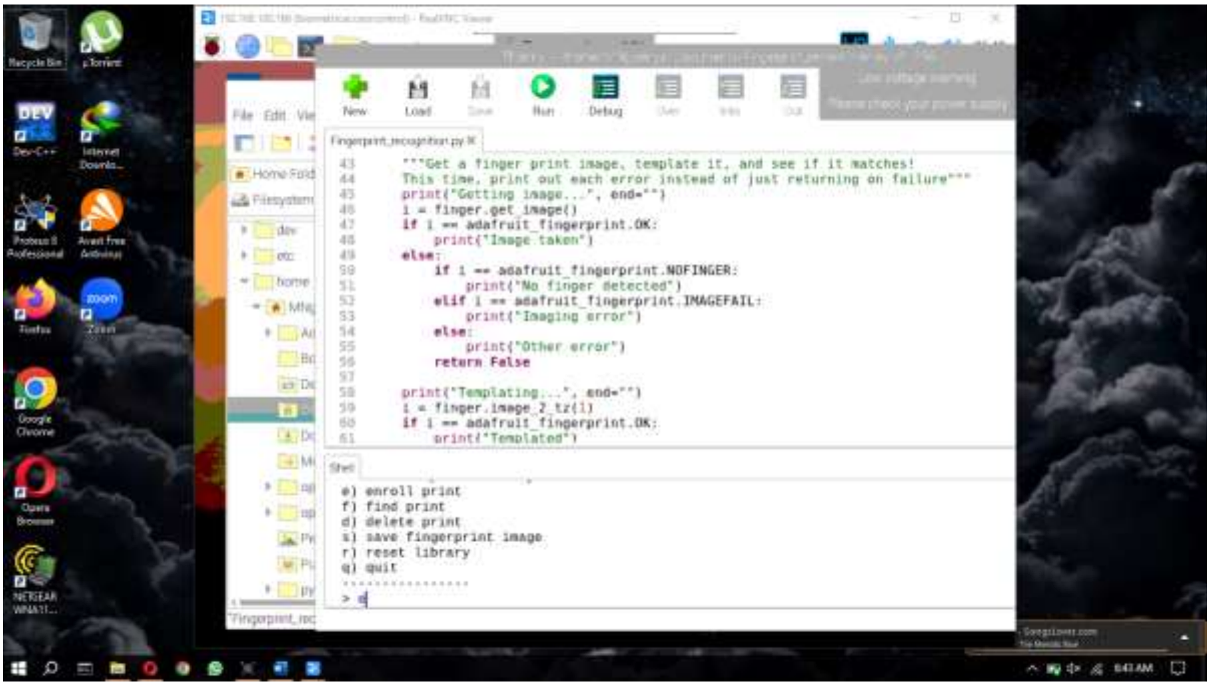
```
pture, format="bgr", use_video_port=True):
face", image)
age_{}.jpg".format(img_counter)
            print("{} written!".format(img_name))
            img_counter += 1

    if k%256 == 27:
```

```
dataset/Engineer Melusi/image_2.jpg written!
dataset/Engineer Melusi/image_3.jpg written!
dataset/Engineer Melusi/image_4.jpg written!
dataset/Engineer Melusi/image_5.jpg written!
dataset/Engineer Melusi/image_6.jpg written!
dataset/Engineer Melusi/image_7.jpg written!
dataset/Engineer Melusi/image_8.jpg written!
```

Face Recognition machine learning

ii. Fingerprint Recognition Training

The engineer began the fingerprint recognition training by running the program to register and store known fingerprints. After known fingerprints have been read and stored the user compiled the programme to check for a known fingerprint and send a signal to open the door once a known print is registered.

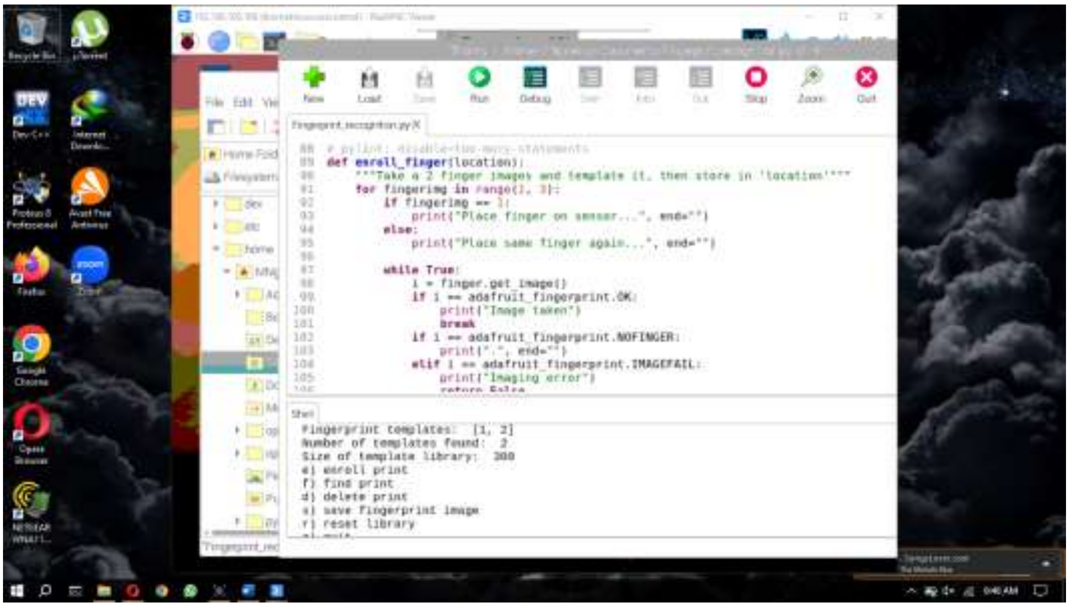


```
    """Get a finger print image, template it, and see if it matches!
    This time, print out each error instead of just returning on failure"""
    print("Getting image...", end="")
    i = finger.get_image()
    if i == adafruit_fingerprint.OK:
        print("Image taken")
    else:
        if i == adafruit_fingerprint.NOFINGER:
            print("No finger detected")
        elif i == adafruit_fingerprint.IMAGEFAIL:
            print("Imaging error")
        else:
            print("Other error")
        return False

    print("Templating...", end="")
    i = finger.image_2_tz(1)
    if i == adafruit_fingerprint.OK:
        print("Templated")
```

```
e) enroll print
f) find print
d) delete print
s) save fingerprint image
r) reset library
q) quit
----------------
> d
```

Fingerprint Recognition Training

Verifying the fingerprint is done by running the same programme and pressing f. Implementing fingerprint verification in Python also makes use of the OpenCV library. The programme implementation includes the necessary code for capturing and processing fingerprint images, as well as the implementation of the matching algorithms.



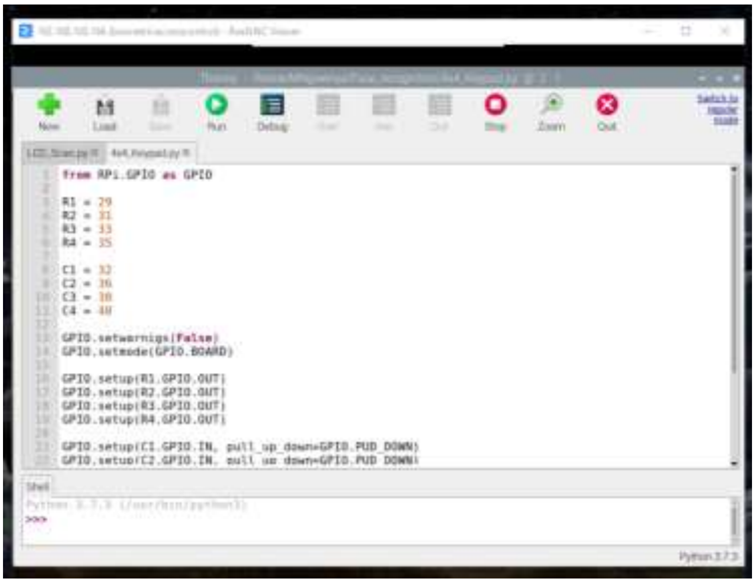
```
# pylint: disable=too-many-statements
def enroll_finger(location):
    """Take a 2 finger images and template it, then store in 'location'"""
    for fingerimg in range(1, 3):
        if fingerimg == 1:
            print("Place finger on sensor...", end="")
        else:
            print("Place same finger again...", end="")

        while True:
            i = finger.get_image()
            if i == adafruit_fingerprint.OK:
                print("Image taken")
                break
            if i == adafruit_fingerprint.NOFINGER:
                print(".", end="")
            elif i == adafruit_fingerprint.IMAGEFAIL:
                print("Imaging error")
                return False
```

```
Fingerprint templates:  [1, 2]
Number of templates found:  2
Size of template library:  200
e) enroll print
f) find print
d) delete print
s) save fingerprint image
r) reset library
q) quit
```

Fingerprint Verification

iii. Keypad Pin Verification

The PIN verification was added as a third authentication for users without biometric verification access. A user is given a PIN as a method of verification until their biometric credentials are features are registered in the system. The PIN, a four-digit combination, is given as a temporary basis and can also be used in extreme conditions when a user has failed to verify using their biometric features. To enter the PIN, a user presses the combination of numbers on the 4x4 keypad. The default PIN that was setup is “1234”.



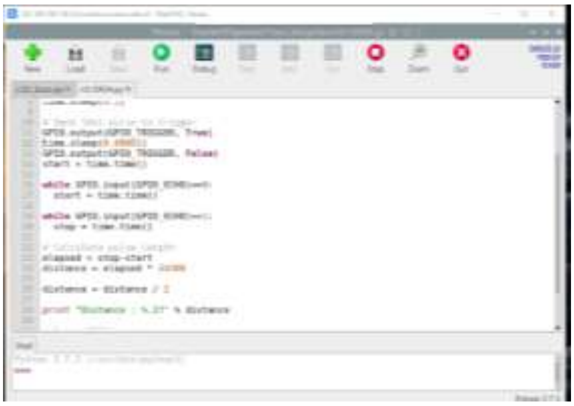
```
from RPi.GPIO as GPIO

R1 = 29
R2 = 31
R3 = 33
R4 = 35

C1 = 32
C2 = 36
C3 = 38
C4 = 40

GPIO.setwarnigs(False)
GPIO.setmode(GPIO.BOARD)

GPIO.setup(R1,GPIO.OUT)
GPIO.setup(R2,GPIO.OUT)
GPIO.setup(R3,GPIO.OUT)
GPIO.setup(R4,GPIO.OUT)

GPIO.setup(C1,GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C2,GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```
Python 3.7.3 (/usr/bin/python3)
>>>
```

Keypad Configuration for PIN

iv. Ultrasonic Range Sensor Configuration

The Ultrasonic range sensor was also configured and set for range sensing using Python. It was set to measure objects in the range of 1- 4 meters from the unit.

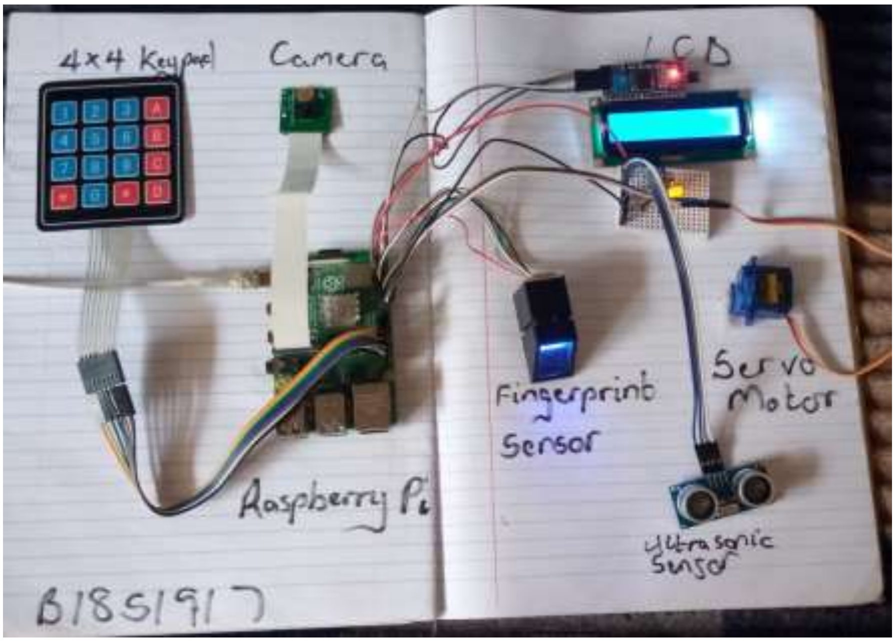


```
# Send 10us pulse to trigger
GPIO.output(GPIO_TRIGGER, True)
time.sleep(0.00001)
GPIO.output(GPIO_TRIGGER, False)
start = time.time()

while GPIO.input(GPIO_ECHO)==0:
  start = time.time()

while GPIO.input(GPIO_ECHO)==1:
  stop = time.time()

# Calculate pulse length
elapsed = stop-start
distance = elapsed * 34300

distance = distance / 2

print "Distance : %.1f" % distance
```

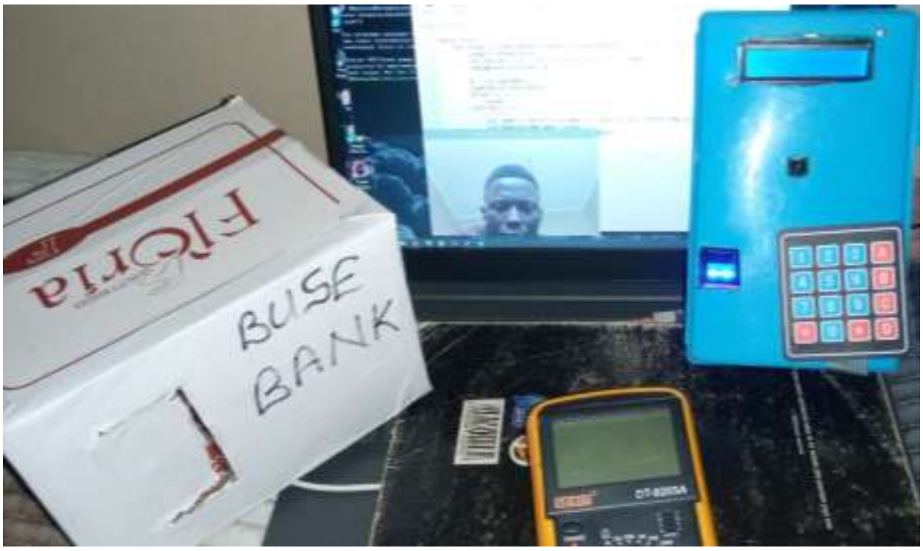
HC-SR04 Configuration

3.6 : Prototype Setup

After the configuration of the different modules was done, the engineer then went on to produce a prototype of the circuit. Due to the limitation in the availability of resources, the magnetic door lock mechanism was replaced by a standard 9g micro-servo motor. The motor which was connected to GPIO12 gave an output of a 90-degree rotation which was attached to the prototype door. The system was placed in a prototype housing unit for protection. An actual housing and mounting rack can be designed based on these and other factors.



Components Layout



3.7 : Conclusion

In this chapter the engineer was able to come up with the basic setup of the components that were required for the biometric access control system to function. These components were then used to create a prototype of the circuit. Each component used was configured mainly using Python Programming language.

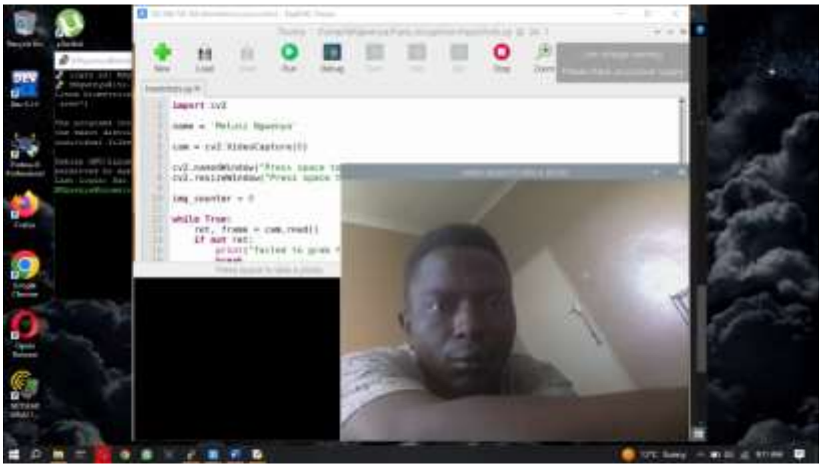
CHAPTER 4: RESULTS

4.1 : Introduction

The final prototype was put to the test and the output results recorded in this chapter. The testing methodology involved collecting data from a sample of users and analysing the system's ability to correctly identify and authenticate them. The system's response to various environmental factors, such as lighting and temperature was also examined and recommendations made on countering the effects. The results of our testing provide valuable insights into the performance of this biometric access control system, which can be used to improve its design and implementation of the final product. Overall, our findings demonstrate the importance of rigorous testing in ensuring the effectiveness of biometric access control systems.

4.2 : Face Recognition Testing

To beginning the testing phase, the engineer began by running the system for facial recognition. It was observed that as soon as the camera was activated and a known face was scanned, the servo motor rotated pulling the door mechanism to open. An unknown face resulted in the LCD displaying a message informing the user to retry scanning.



```
import cv2

name = 'Melusi Ngwenya'

cam = cv2.VideoCapture(0)

cv2.namedWindow("Press space to
cv2.resizeWindow("Press space t

img_counter = 0

while True:
    ret, frame = cam.read()
    if not ret:
        print("failed to grab "
        break
```

Face Recognition

4.3 : Fingerprint Verification Testing

The second phase of testing was conducted on the fingerprint verification and recognition programme. When a registered fingerprint was scanned, the servo motor rotated pulling the door mechanism to open. An unknown fingerprint resulted in the user being prompted to retry scanning their fingerprint.

4.4 : PIN Authorization Testing

The last authentication to be done was the PIN authorisation access. This also gave a rotation of the servo motor indicating successful authentication and access approval. A wrong PIN resulted in the user being prompted to try again entering the 4 digit pin.

4.5 : Ultrasonic range sensor Testing

Last but not least, the testing of the range sensor for various distances of a human approaching the locking system was done. The best results were noted when a user was between 1 and 3 meters away from the system. When the user was in this range this is when the camera was activated for face recognition.

5.2 Aims and Objectives Realization

Research Objective	**Objective Met**
Ensure safety and security of the organisation’s premises.	**YES**
IoT support for the system to record and monitor access	**YES**
To ensure fingerprint authentication on the system	**YES**
To ensure facial recognition on the system	**YES**

4.6 : Conclusion

The results from this testing phase show that the system was able to accurately verify the identity of authorized personnel, while denying access to unauthorized individuals. The system's high accuracy and speed make it a promising solution for organizations looking to enhance their security measures. However, further testing is required to evaluate the system's performance under different environmental conditions and against potential attacks. Overall, our findings suggest that biometric access control systems have the potential to significantly improve security in various settings.

CHAPTER 5: RECOMMENDATIONS

5.1 : Introduction

In this Chapter, a presentation of the recommendations for improving the effectiveness of biometric access control systems was conducted. The challenges faced during the course of the project development were also listed.

5.2 : Recommendations

- ✓ Firstly, it is crucial to ensure that the biometric system is well-designed and implemented, with a high level of accuracy and reliability. This can be achieved through rigorous testing and evaluation, as well as ongoing monitoring and maintenance.

- ✓ Secondly, it is important to consider the privacy and security implications of biometric data. Access control systems should be designed with privacy in mind, and data should be stored securely and protected from unauthorized access.

- ✓ Thirdly, it is recommended that biometric access control systems are integrated with other security measures, such as video surveillance and intrusion detection systems. This can help to provide a more comprehensive and effective security solution.

- ✓ Finally, it is important to provide ongoing training and education to users of biometric access control systems. This can help to ensure that users are aware of the system's capabilities and limitations, and can use it effectively and appropriately.

In conclusion, by following these recommendations, biometric access control systems can be made more effective and secure, providing a valuable tool for enhancing access control to the bank’s premises.

5.3 : Challenges Faced

The challenges that were faced in the course of the design project are;

i. Frequent power cuts which hindered the smooth flow of the project development and prototype design and testing.

ii. Limited availability of a laboratory hence determination of available resources was nearly impossible.

iii. Limited access to internet for research purposes.

The recommended solutions that were given by the engineer include but are not limited to;

- ✓ Having backup power sources to avoid disruptions caused by electricity power cuts.
- ✓ Selecting a stand-alone laboratory for the design of the prototypes.
- ✓ Gathering all the required resources and data for training the systems and machine learning in the early stages of the design.
- ✓ Perform regular tests on the output during and after each development of a feature
- ✓ Creating a regular checked and updated Log book to track progress of the design project.

The engineer then went on to submit the design prototype to the security department for further in house training and testing of the access control system. This stage would then lead to the development of the actual system after approval by the bank’s Head of Security.

APPENDIX

i. Face recognition code

```
import face_recognition.api as face_recognition
import multiprocessing
import itertools
import sys
import PIL.Image
import numpy as np


def scan_known_people(known_people_folder):
    known_names = []
    known_face_encodings = []


    for file in image_files_in_folder(known_people_folder):
        basename = os.path.splitext(os.path.basename(file))[0]
        img = face_recognition.load_image_file(file)
        encodings = face_recognition.face_encodings(img)


        if len(encodings) > 1:
            click.echo("WARNING: More than one face found in {}. Only considering the first face.".format(file))


        if len(encodings) == 0:
            click.echo("WARNING: No faces found in {}. Ignoring file.".format(file))
        else:
            known_names.append(basename)
            known_face_encodings.append(encodings[0])

    return known_names, known_face_encodings


def print_result(filename, name, distance, show_distance=False):
    if show_distance:
        print("{},{},{}".format(filename, name, distance))
    else:
        print("{},{}".format(filename, name))


def test_image(image_to_check, known_names, known_face_encodings, tolerance=0.6,
show_distance=False):
    unknown_image = face_recognition.load_image_file(image_to_check)
    if max(unknown_image.shape) > 1600:
        pil_img = PIL.Image.fromarray(unknown_image)
        pil_img.thumbnail((1600, 1600), PIL.Image.LANCZOS)
        unknown_image = np.array(pil_img)

    unknown_encodings = face_recognition.face_encodings(unknown_image)

    for unknown_encoding in unknown_encodings:
        distances = face_recognition.face_distance(known_face_encodings,
unknown_encoding)
        result = list(distances <= tolerance)

        if True in result:
            [print_result(image_to_check, name, distance, show_distance) for is_match, name, distance in zip(result, known_names, distances) if is_match]
        else:
            print_result(image_to_check, "unknown_person", None, show_distance)

    if not unknown_encodings:
        # print out fact that no faces were found in image
        print_result(image_to_check, "no_persons_found", None, show_distance)

def image_files_in_folder(folder):
    return [os.path.join(folder, f) for f in os.listdir(folder) if re.match(r'.*\.(jpg|jpeg|png)', f, flags=re.I)]

def process_images_in_process_pool(images_to_check, known_names, known_face_encodings, number_of_cpus, tolerance, show_distance):
    if number_of_cpus == -1:
        processes = None
    else:
        processes = number_of_cpus
    context = multiprocessing
    if "forkserver" in multiprocessing.get_all_start_methods():
        context = multiprocessing.get_context("forkserver")

    pool = context.Pool(processes=processes)

    function_parameters = zip(
        images_to_check,
        itertools.repeat(known_names),
        itertools.repeat(known_face_encodings),
        itertools.repeat(tolerance),
        itertools.repeat(show_distance)
    )
    pool.starmap(test_image, function_parameters)
@click.command()
@click.argument('known_people_folder')
@click.argument('image_to_check')
@click.option('--cpus', default=1, help='number of CPU cores to use in parallel (can speed up processing lots of images). -1 means "use all in system"')
@click.option('--tolerance', default=0.6, help='Tolerance for face comparisons. Default is 0.6. Lower this if you get multiple matches for the same person.')
@click.option('--show-distance', default=False, type=bool, help='Output face distance. Useful for tweaking tolerance setting.')
def main(known_people_folder, image_to_check, cpus, tolerance, show_distance):
    known_names, known_face_encodings = scan_known_people(known_people_folder)
    if (sys.version_info < (3, 4)) and cpus != 1:
        click.echo("WARNING: Multi-processing support requires Python 3.4 or greater. Falling back to single-threaded processing!")
        cpus = 1

    if os.path.isdir(image_to_check):
        if cpus == 1:
            [test_image(image_file, known_names, known_face_encodings, tolerance, show_distance) for image_file in image_files_in_folder(image_to_check)]
        else:
            process_images_in_process_pool(image_files_in_folder(image_to_check), known_names, known_face_encodings, cpus, tolerance, show_distance)
    else:
        test_image(image_to_check, known_names, known_face_encodings, tolerance, show_distance)


if __name__ == "__main__":
    main()
```

ii. Fingerprint verification code

```
import hashlib
from pyfingerprint.pyfingerprint import PyFingerprint
from pyfingerprint.pyfingerprint import FINGERPRINT_CHARBUFFER1

#Servo Motor is attached to the GPIO numbered below
Pin = 21

import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(Pin, GPIO.OUT)

## Search for a finger
```

```
##

## Tries to initialize the sensor
try:
    f = PyFingerprint('/dev/ttyS0', 57600, 0xFFFFFFFF, 0x00000000)

    if ( f.verifyPassword() == False ):
        raise ValueError('The given fingerprint sensor password is wrong!')

except Exception as e:
    print('The fingerprint sensor could not be initialized!')
    print('Exception message: ' + str(e))
    exit(1)
print('Currently used templates: ' + str(f.getTemplateCount()) +'/'+
str(f.getStorageCapacity()))

## Tries to search the finger and calculate hash
try:
    print('Waiting for finger...')

    ## Wait that finger is read
    while ( f.readImage() == False ):
        pass

    f.convertImage(FINGERPRINT_CHARBUFFER1)
    result = f.searchTemplate()

    positionNumber = result[0]
    accuracyScore = result[1]

    if ( positionNumber == -1 ):
        print('No match found!')
        exit(0)

    else:
        print('Found template at position #' + str(positionNumber))
        print('The accuracy score is: ' + str(accuracyScore))

        GPIO.output(Pin, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(Pin, GPIO.LOW)
    f.loadTemplate(positionNumber, FINGERPRINT_CHARBUFFER1)
    characterics = str(f.downloadCharacteristics(FINGERPRINT_CHARBUFFER1)).encode('utf-8')

       print('SHA-2 hash of template: ' + hashlib.sha256(characterics).hexdigest())

except Exception as e:
    print('Operation failed!')
    print('Exception message: ' + str(e))
    exit(1)
```

iii. HC-SR04 Range Sensor Code

```
# Set pins as output and input
GPIO.setup(GPIO_TRIGGER,GPIO.OUT)  # Trigger
GPIO.setup(GPIO_ECHO,GPIO.IN)      # Echo

# Set trigger to False (Low)
GPIO.output(GPIO_TRIGGER, False)

time.sleep(0.5)

# Send 10us pulse to trigger
GPIO.output(GPIO_TRIGGER, True)
time.sleep(0.00001)
GPIO.output(GPIO_TRIGGER, False)
start = time.time()

while GPIO.input(GPIO_ECHO)==0:
  start = time.time()

while GPIO.input(GPIO_ECHO)==1:
  stop = time.time()
```

```
# Calculate pulse length
elapsed = stop-start
distance = elapsed * 34300

distance = distance / 2

print "Distance : %.1f" % distance

# Reset GPIO settings
GPIO.cleanup()
```

iv. Keypad Pin Code

```
import RPi.GPIO as GPIO
import time

L1 = 5
L2 = 6
L3 = 13
L4 = 19

# These are the four columns
C1 = 12
C2 = 16
C3 = 20
C4 = 21
```

```
keypadPressed = -1

secretCode = "1234"
input = ""

# Setup GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

GPIO.setup(L1, GPIO.OUT)
GPIO.setup(L2, GPIO.OUT)
GPIO.setup(L3, GPIO.OUT)
GPIO.setup(L4, GPIO.OUT)

# Use the internal pull-down resistors
GPIO.setup(C1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def keypadCallback(channel):
    global keypadPressed
    if keypadPressed == -1:
        keypadPressed = channel

# Detect the rising edges on the column lines of the
# keypad. This way, we can detect if the user presses
# a button when we send a pulse.
GPIO.add_event_detect(C1, GPIO.RISING, callback=keypadCallback)
GPIO.add_event_detect(C2, GPIO.RISING, callback=keypadCallback)
GPIO.add_event_detect(C3, GPIO.RISING, callback=keypadCallback)
GPIO.add_event_detect(C4, GPIO.RISING, callback=keypadCallback)

# Sets all lines to a specific state. This is a helper
# for detecting when the user releases a button
def setAllLines(state):
    GPIO.output(L1, state)
    GPIO.output(L2, state)
    GPIO.output(L3, state)
    GPIO.output(L4, state)

def checkSpecialKeys():
    global input
    pressed = False

    GPIO.output(L3, GPIO.HIGH)

    if (GPIO.input(C4) == 1):
        print("Input reset!");
        pressed = True

    GPIO.output(L3, GPIO.LOW)
    GPIO.output(L1, GPIO.HIGH)
    if (not pressed and GPIO.input(C4) == 1):
        if input == secretCode:
            print("Code correct!")
            # TODO: Unlock door
        else:
            print("Incorrect code!")
            # TODO: Sound an alarm, send an email, etc.
        pressed = True

    GPIO.output(L3, GPIO.LOW)


    if pressed:
        input = ""


    return pressed
def readLine(line, characters):
    global input
    # We have to send a pulse on each line to
    # detect button presses
    GPIO.output(line, GPIO.HIGH)
    if(GPIO.input(C1) == 1):
        input = input + characters[0]
    if(GPIO.input(C2) == 1):
        input = input + characters[1]
    if(GPIO.input(C3) == 1):
        input = input + characters[2]
    if(GPIO.input(C4) == 1):
        input = input + characters[3]
    GPIO.output(line, GPIO.LOW)

try:
    while True:
        if keypadPressed != -1:
            setAllLines(GPIO.HIGH)
            if GPIO.input(keypadPressed) == 0:
                keypadPressed = -1
            else:
                time.sleep(0.1)
        # Otherwise, just read the input
        else:
            if not checkSpecialKeys():
                readLine(L1, ["1","2","3","A"])
                readLine(L2, ["4","5","6","B"])
                readLine(L3, ["7","8","9","C"])
                readLine(L4, ["*","0","#","D"])
                time.sleep(0.1)
            else:
                time.sleep(0.1)
except KeyboardInterrupt:
    print("\nApplication stopped!")
```

v. LCD Display code

```
lcd=LiquidCrystal_I2C.lcd()

lcd.clear()

lcd.display("Scannig....",1,0)

sleep(1)

lcd.clear()

for j in range(1,3):

    for i in range(16):

      lcd.display("*",j,i)

  sleep(0.1)

lcd.clear()

while True:

  try:

  lcd.display("Please Try Scanning Again",1,0)

  lcd.display("want to display",2,0)

  sleep(1)

lcd.display(input("Success"),1,0) sleep(2)

lcd.clear()

except KeyboardInterrupt: break
```

REFERENCES

1. Ross, A., Jain, A. K., & Nandakumar, K. (2010). Handbook of biometrics. Springer.

2. Rattani, A., & Derakhshani, R. (2019). Biometric authentication using human gait analysis: A review. IEEE Access, 7, 141509-141526.

3. J. Yan, S. Smith, and Q. Li. "Biometric authentication: a machine learning approach." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 4, pp. 593-606, 2007.

4. Onyan AO, and Enalume KO. “Property Security Using a Biometric Based Door Lock System” Journal of Biostatistics and Biometric Applications, vol. 3, Issue no. 3, pp. 1-2, 2018.

5. Ratha, N. K., Connell, J. H., & Bolle, R. M. (2001). Enhancing security and privacy in biometrics-based authentication systems. IBM systems Journal, 40(3), 614-634.

6. Li, S. Z., & Jain, A. K. (Eds.). (2005). Handbook of face recognition. Springer Science & Business Media.

7. Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2009). Handbook of fingerprint recognition. Springer Science & Business Media.

8. M. A. S. Zulkernine, S. S. Iyengar, and S. Venkatesan, “Biometric access control systems: A survey,” ACM Computing Surveys, vol. 39, no. 4, pp. 1–42, 2007.

9. J. Y. Chen, C. Y. Huang, and C. H. Tsai, “A survey of biometric authentication techniques,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 6, pp. 1067–1080, 2007.

10. S. S. Iyengar and S. Venkatesan, “Biometric access control systems: Security and privacy issues,” IEEE Security & Privacy, vol. 3, no. 5, pp. 58–63, 2005.

11. M. A. Karim, A. K. Roy, and S. A. S. M. Zaidi, “Biometric access control systems: Issues and challenges,” IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 1745–1763, 2013.

12. Hung, Chi-Huang & Bai, Ying-Wen & Ren, Je-Hong. “Design and implementation of a door lock control based on a near field communication of a smartphone”. 45-46. 10.1109/ICCE-TW, 2015. Retrieved from https://ieeexplore.ieee.org/xpl/conhome/7170113/proceeding

13. GeeksforGeeks: https://www.geeksforgeeks.org/python-programming-language/

14. OpenCV: https://opencv.org/

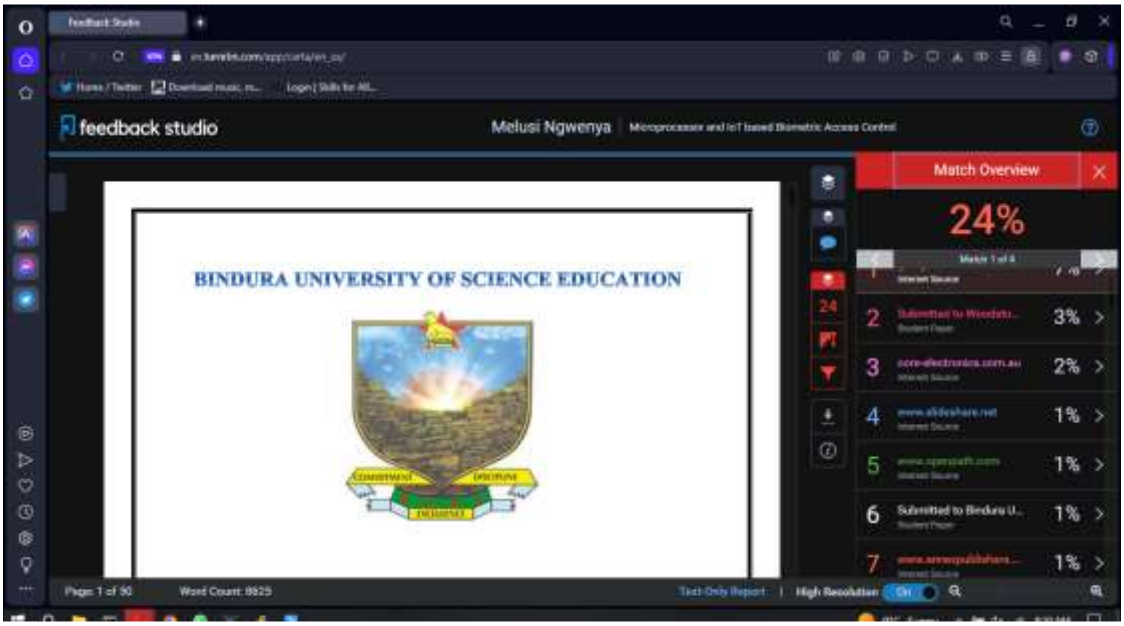
15. https://www.raspberrypi-spy.co.uk/2012/12/ultrasonic-distance-measurement-using-python-part-1

16. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

17. https://circuitdigest.com/

18. https://components101.com/

Turnitin Score



feedback studio

Melusi Ngwenya | Microprocessor and IoT based Biometric Access Control

BINDURA UNIVERSITY OF SCIENCE EDUCATION

Match Overview

24%

Match 1 of 8

1	Internet Source	
2	Submitted to Woodstr... Student Paper	3%
3	core-electronics.com.au Internet Source	2%
4	www.slideshare.net Internet Source	1%
5	www.openpaft.com Internet Source	1%
6	Submitted to Bindura U... Student Paper	1%
7	www.amerpublishers... Internet Source	1%

Page: 1 of 90 Word Count: 8825

Text-Only Report | High Resolution On