**CHAPTER 1**

# **INTRODUCTION**

## **1.1 Background**

Access control systems are electronic systems that are designed to control security through a network. Access Control System recognizes, authenticates and authorizes entry of an individual into a specific region or space, thereby giving complete protection and ensuring security of area accessed.

Access control systems within a building may be linked or standardized based on the size of the organization and the varying levels of security. It is suitable for homes, offices and other access control applications. Access control systems were typically administered in a central location. Besides electronic door locks, there are access control panel models, magnetic door locks with uninterrupted power supplies. An administrator grants access to a person with the settings built inside the administration software and a key which opens all of the doors in a building in a door access control system and these update the information within the system and each user being given a unique identity.

Preventing unauthorized access to a region or space has evolved from locks and keys to sophisticated electronic equipment, but the goal of maintaining the integrity of an area or space remains the same. One of the earliest entries into the security access field, key pads allowed employees and other authorized users to enter a personal identification number (PIN) that matched data in an electronic device. The system provided minimal security, but it was more efficient than locks and keys. Forgetting an assigned PIN or sharing it with others limited the effectiveness of the system.

An improvement on using PINs for access control occurred with the development of a card that used Wigand technology to read a magnetic strip affixed to a plastic card. Commonly called “key cards” or “card keys”, the system has some of the same defects that made the keypad system ineffective. Swipe readers require a user to slide a card through a long narrow slot, and they are frequently found in retail stores. An insert reader accepts a card in a slot and allows the holder to authorize transactions. Proximity readers can interpret information on the face of a card when it is within a specified distance.

These days, areas can be secured using biometric information of the user. The term “biometric” simply refers to body measurement and calculations. Biometric authentication is used as a form of identification and access control. Biometrics can range from fingerprint scanning to palm print, face recognition to iris recognition. The biometric device is a security authentication device. Such a device uses automated methods of verifying or recognizing the specified characteristics of the individual.

An advantage of the biometric security is that biometric data cannot be easily stolen and hacking is extremely complicated, especially iris recognition. Also, you cannot forget your biometric information, even if you try to.

Modern devices and security methods all try to integrate the pin lock and a biometric security access control. Usually the PIN/password being the backup for any error in the system.

Some systems can have as much as 4 levels of security with over half being biometric security levels for increased security. In this research, a PIN lock and biometric (fingerprint scanning) are used, hence providing 2 levels of security.

## **1.2 Problem Statement**

This project enables physical security in places of use by identifying every individual accessing the secured area or space using the PIN lock and biometric (fingerprint) security. The uniqueness of this project includes using different layers of security measures to ensure maximum security and proffering solutions to correct the flaws of different versions.

## **1.3 Objectives**

To achieve the desired access control system, the following objectives were considered:

a. To implement a system that grants access only to authorized persons using biometric security and PIN lock;

b. To improve efficiency of security in specific areas and space to ensure the safety of property;

c. To produce an audio output to prompt user of the system on how to make use of the system;

d. To produce a light indication to show the state of the system;

e. To produce a visual output on a screen to prompt the user on how to make use of the system. Also, the visual output shows the user the state of the system.

**1.4 Aims**

To achieve the access control system, the aims include:

a. To implement an automated access control system using a keypad and fingerprint sensor;

b. To produce an audio output using a speaker to prompt user on steps to be taken when using system;

c. To produce light indication using Light Emitting Diodes of various colors just like the traffic light;

d. To produce a visual output using a Liquid Crystal Display (LCD) which prompts the user on necessary steps to take when using the system and states the current state of the system.

## **1.5 Scope**

This is a compact prototype for the design of an Access Control System using fingerprint sensor and PIN lock. This system produces audio output, a display on a screen(LCD) and light indication. The versatility of this system can be seen in everyday life ranging our smartphones to Automated Teller Machines and even in automated doors.

**1.6 Organization of the Report**

The remainder of the report is organized as follows: chapter 2 presents the literature review, chapter 3 presents the methodology, chapter 4 presents the results and chapter 5 concludes the report.

**CHAPTER 2**

# **LITERATURE REVIEW**

## **2.1 The Fingerprint**

Fingerprinting is one form of biometrics, a science that uses people's physical characteristics to identify them. Fingerprints are ideal for this purpose because they're inexpensive to collect and analyze, and they never change, even as people age.

Although hands and feet have many ridged areas ­that could be used for identification, fingerprints became a popular form of biometrics because they are easy to classify and sort. They're also accessible.

Fingerprints are made of an arrangement of ridges, called friction ridges, as seen in Figure2.1. Each ridge contains pores, which are attached to sweat glands under the skin. You leave fingerprints on glasses, tables and just about anything else you touch because of this sweat.

All the ridges of fingerprints form patterns called loops, whorls or arches:

a. Loops begin on one side of the finger, curve around or upward, and exit the other side. There are two types of loops: radial loops and ulnar loops. Radial loops slope towards the thumb while ulnar loops slope towards the little finger

b. Whorls form a circular or spiral pattern

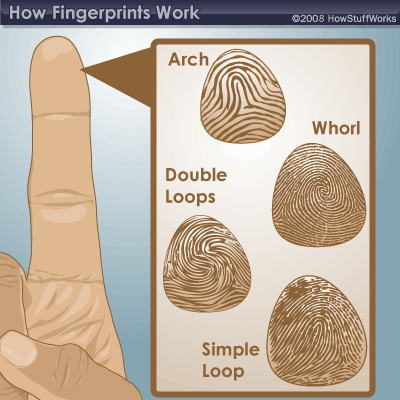
c. Archesslope upward and then down, like very narrow mountains.

Scientists look at the arrangement, shape, size and number of lines in these fingerprint patterns to distinguish one from another. They also analyze very tiny characteristics called minutiae, which can't be seen with the naked eye.

If fingerprints are so unique and subtle, how are they recorded accurately? Dactyloscopy! This is the art of fingerprinting.

Deliberate impression of fingerprints may be formed by or other substances transferred from the peaks of friction ridges on the skin to a relatively smooth surface such as a fingerprint card. Fingerprint records normally contains impressions from the pad on the last joint of fingers and thumbs, although fingerprint cards also typically records portion of lower joint areas of the fingers.

Human finger prints are detailed, nearly unique, difficult to alter, and durable over the life of an individual, making them suitable as long-term makers of human identity. Fingerprints differ from finger to finger also. They may be employed by police or other authorities to identify individuals who wish to conceal their identity, or to identify people who are incapacitated or deceased and thus unable to identify themselves, as in the aftermath of a natural disaster. Fingerprint analysis, in use since the early 20th century has led to many crimes being solved. This means that many criminals consider gloves essential. In 2015, the identification of sex by use of fingerprint has been reported.



**Figure 2.1:** A diagram of the various kinds of fingerprints

## **2.2 Fingerprint Sensor (Adafruit)**

 A **s**ensor is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor[4]. The fingerprint sensor used in this project is the adafruit fingerprint sensor shown in Figure 2.2. It has the specifications listed in Table 2.1;

**Figure2.2**: Adafruit Fingerprint Sensor[4]

|  |  |
| --- | --- |
| **Specification** | **Rating** |
| Supply voltage | 3.6 - 6.0VDC |
| Peak current | 150mA max |
| Working current | 120mA |
| Window area | 14mm x 18mm |
| Signature file | 256 bytes |
| Template file | 512 bytes |
| Full Dimensions | 56 x 20 x 21.5mm |
| Exposed Dimensions | 21mm x 21mm x 21mm triangular |
| Weight | 20 grams |
| Working temperature rating | 20C to +50C |
| Working humidity | 40%-85% RH |
| Interface | TTL Serial |
| False Acceptance Rate | <0.001% (Security level 3) |
| False Reject Rate | <1.0% (Security level 3) |
| Storage capacity | 162 templates |
| Safety ratings | (1-5 low to high safety) |
| Baud rate | 9600, 19200, 28800, 38400, 57600 (default) |

**Table 2.1**: Adafruit Fingerprint Sensor Specification[4].

## 

## **2.3Arduino Mega 2560**

The **Arduino Mega 2560** is a microcontroller board based on the ATMEGA 2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board in Figure 2.3 is compatible with most shields designed for the Uno and the former board

 **Figure 2.3**: Arduino Mega 2560

## **2.4 Arduino Uno**

The Arduino Uno is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board shown in Figure 2.4 is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable). It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts. It is also like the [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) and Leonardo. The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "[uno](https://en.wiktionary.org/wiki/uno" \o "wikt:uno)" means "one" in [Italian](https://en.wikipedia.org/wiki/Italian_language) and was chosen to mark the initial release of the [Arduino Software](https://en.wikipedia.org/wiki/Arduino_Software). The Uno board is the first in a series of USB-based Arduino boards, and it and version 1.0 of the Arduino [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment) were the reference versions of Arduino, now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a [bootloader](https://en.wikipedia.org/wiki/Bootloader) that allows uploading new code to it without the use of an external hardware programmer. [5].

The specification and ratings for Arduino Uno is shown in Table 2.2

**Table 2.2:** Arduino Uno Specification

|  |  |
| --- | --- |
| **Specification** | **Rating** |
| [Microcontroller](https://en.wikipedia.org/wiki/Microcontroller): | [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) [[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7) |
| Operating Voltage: | 5 Volts |
| Input Voltage: | 7 to 20 Volts |
| Digital I/O Pins: | 14 (of which 6 provide PWM output) |
| Analog Input Pins: | 6 |
| DC Current per I/O Pin: | 20 mA |
| DC Current for 3.3V Pin: | 50 mA |
| [Flash Memory](https://en.wikipedia.org/wiki/Flash_Memory): | 32 KB of which 0.5 KB used by [bootloader](https://en.wikipedia.org/wiki/Booting#BOOT-LOADER) |
| [SRAM](https://en.wikipedia.org/wiki/Static_random-access_memory): | 2 KB |
| [EEPROM](https://en.wikipedia.org/wiki/EEPROM): | 1 KB |
| Clock Speed: | 16 MHz |
| Length: | 68.6 mm |
| Width: | 53.4 mm |
| Weight: | 25 g |

Arduino Uno has 14 pins with 7 being general pins. In the Table 2.3, the general pin functions are discussed.

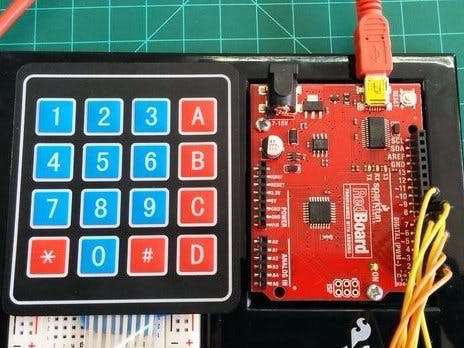
**Table 2.3:** General pin functions of Arduino Uno[5]

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Pin Name** | | **Function** |
| 1 | | LED | There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it's off. |
| 2 | | VIN | The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. |
| 3 | | 5V | This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board. |
| 4 | | 3V3 | A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA. |
| 5 | | GND | Ground pins. |
| 6 | | IOREF | This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V. |
| 7 | | Reset | Typically used to add a reset button to shields which block the one on the board. |

 **Figure 2.4:** Arduino Uno[5]

**2.5 Keypad**

Keypads are used in all types of devices, including cell phones, fax machines, microwaves, ovens, door locks, etc. They are practically everywhere. Many electronic devices use keypads for user input. For this project, the type of keypad used is a 4X4 matrix keypad shown in Figure 2.5.



**Figure 2.5**: Keypad hardware

## **2.6 Battery**

A battery is a device consisting of one or more [electrochemical cells](https://en.wikipedia.org/wiki/Electrochemical_cell) with external connections provided to power electrical devices such as [flashlights](https://en.wikipedia.org/wiki/Flashlight), [mobile phones](https://en.wikipedia.org/wiki/Mobile_phone), and [electric cars](https://en.wikipedia.org/wiki/Electric_car). Batteries convert [chemical energy](https://en.wikipedia.org/wiki/Chemical_energy) directly to [electrical energy](https://en.wikipedia.org/wiki/Electrical_energy).

 When a battery is supplying [electric power](https://en.wikipedia.org/wiki/Electric_power), its positive terminal is the [cathode](https://en.wikipedia.org/wiki/Cathode) and its negative terminal is the [anode](https://en.wikipedia.org/wiki/Anode), as shown in Figure 2.6(b). The terminal marked negative is the source of electrons that will flow through an external electric circuit to the positive terminal. When a battery is connected to an external electric load, a [redox](https://en.wikipedia.org/wiki/Redox) reaction converts high-energy reactants to lower-energy products, and the [free-energy](https://en.wikipedia.org/wiki/Gibbs_free_energy) difference is delivered to the external circuit as electrical energy. Historically the term "battery" specifically referred to a device composed of multiple cells, however the usage has evolved to include devices composed of a single cell.

Batteries are classified into primary and secondary forms:

a. Primary batteries are used until exhausted of energy then discarded. Their chemical reactions are not reversible, so they cannot be recharged. When the supply of reactants in the battery is exhausted, the battery stops producing current and is useless. An example of primary battery can be seen in Figure 2.6(a).

b. Secondary batteries can be recharged; that is, they can have their chemical reactions reversed by applying [electric current](https://en.wikipedia.org/wiki/Electric_current) to the cell. This regenerates the original chemical reactants, so they can be used, recharged, and used again multiple times.

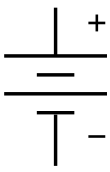
Some types of primary batteries used, for example, for [telegraph](https://en.wikipedia.org/wiki/Telegraphy) circuits, were restored to operation by replacing the electrodes. Secondary batteries are not indefinitely rechargeable due to dissipation of the active materials, loss of electrolyte and internal corrosion.

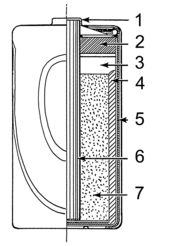
A battery consists of some number of [voltaic cells](https://en.wikipedia.org/wiki/Voltaic_cell). Each cell consists of two [half-cells](https://en.wikipedia.org/wiki/Half-cell) connected in series by a conductive [electrolyte](https://en.wikipedia.org/wiki/Electrolyte) containing metal cations. One half-cell includes electrolyte and the negative electrode, the electrode to which [anions](https://en.wikipedia.org/wiki/Ion#Anions_and_cations) (negatively charged ions) migrate; the other half-cell includes electrolyte and the positive electrode, to which [cations](https://en.wikipedia.org/wiki/Ion#Anions_and_cations) (positively charged [ions](https://en.wikipedia.org/wiki/Ion)) migrate, as shown in Figure2.6(c). Cations are reduced (electrons are added) at the cathode, while metal atoms are oxidized (electrons are removed) at the anode.Some cells use different electrolytes for each half-cell; then a separator is used to prevent mixing of the electrolytes while allowing ions to flow between half-cells to complete the electrical circuit.

Each half-cell has an [electromotive force](https://en.wikipedia.org/wiki/Electromotive_force) (emf, measured in volts) relative to a [standard](https://en.wikipedia.org/wiki/Standard_hydrogen_electrode). The net emf of the cell is the difference between the emfs of its half-cells. Thus, if the electrodes have emfs {\displaystyle {\mathcal {E}}\_{1}}and {\displaystyle {\mathcal {E}}\_{2}}, then the net emf is {\displaystyle {\mathcal {E}}\_{2}-{\mathcal {E}}\_{1}}; in other words, the net emf is the difference between the [reduction potentials](https://en.wikipedia.org/wiki/Reduction_potential) of the [half-reactions](https://en.wikipedia.org/wiki/Half-reaction)[6].

The electrical driving force or {\displaystyle \displaystyle {\Delta V\_{bat}}}across the [terminals](https://en.wikipedia.org/wiki/Battery_terminal) of a cell is known as the terminal voltage (difference) and is measured in [volts](https://en.wikipedia.org/wiki/Volt). The terminal voltage of a cell that is neither charging nor discharging is called the [open-circuit voltage](https://en.wikipedia.org/wiki/Open-circuit_voltage) and equals the emf of the cell. Because of internal resistance, the terminal voltage of a cell that is discharging is smaller in magnitude than the open-circuit voltage and the terminal voltage of a cell that is charging exceeds the open-circuit voltage. An ideal cell has negligible internal resistance, so it would maintain a constant terminal voltage of {\displaystyle {\mathcal {E}}}until exhausted, then dropping to zero.

**Figure 2.6(a)**: 9volt cell

**Figure 2.6(b):**  Electronic Representation of a battery[6]

**Figure 2.6(c)** [6]**:** Line art drawing of a dry cell:   
1. brass cap, 2. plastic seal, 3. expansion space, 4. porous cardboard, 5. zinc can, 6. carbon rod, 7. chemical mixture

## **2.7 Light Emitting Diode**

A light-emitting diode (LED) is a [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) [light source](https://en.wikipedia.org/wiki/Light_source) that emits light when [current](https://en.wikipedia.org/wiki/Electric_current) flows through it. [Electrons](https://en.wikipedia.org/wiki/Electron) in the semiconductor recombine with [electron holes](https://en.wikipedia.org/wiki/Electron_hole), releasing energy in the form of [photons](https://en.wikipedia.org/wiki/Photon). The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the [band gap](https://en.wikipedia.org/wiki/Band_gap) of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device[2].

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared light. Infrared LEDs are used in [remote-control](https://en.wikipedia.org/wiki/Remote-control) circuits, such as those used with a wide variety of consumer electronics. The first visible-light LEDs were of low intensity and limited to red. Modern LEDs are available across the [visible](https://en.wikipedia.org/wiki/Visible_spectrum), [ultraviolet](https://en.wikipedia.org/wiki/Ultraviolet), and [infrared](https://en.wikipedia.org/wiki/Infrared) wavelengths, with high light output.

Early LEDs were often used as indicator lamps, replacing small incandescent bulbs, and in [seven-segment displays](https://en.wikipedia.org/wiki/Seven-segment_display). Recent developments have produced high-output white light LEDs suitable for room and outdoor area lighting. LEDs have led to new displays and sensors, while their high switching rates are useful in advanced communications technology.

LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. LEDs are used in applications as diverse as [aviation lighting](https://en.wikipedia.org/wiki/Navigation_light#Aviation_navigation_lights), [automotive headlamps](https://en.wikipedia.org/wiki/Automotive_lighting#Light_emitting_diodes_(LED)), advertising, [general lighting](https://en.wikipedia.org/wiki/Lighting), [traffic signals](https://en.wikipedia.org/wiki/Traffic_signal), camera flashes, lighted wallpaper, [plant growing light](https://en.wikipedia.org/wiki/Grow_light) and medical devices.

Unlike a [laser](https://en.wikipedia.org/wiki/Laser), the color of light emitted from an LED is neither coherent nor monochromatic, but the spectrum is narrow with respect to human vision, and functionally monochromatic[2].

The internal structure of an LED can be seen in Figure 2.7(b).

## **2.7(a) Applications of LEDS**[2]

LED uses fall into four major categories:

i. Visual signals where light goes directly from the source to the human eye, to convey a message or meaning such as in traffic lights as shown in Figure 2.7(a).

ii. [Illumination](https://en.wikipedia.org/wiki/Lighting) where light is reflected from objects to give visual response of these objects

iii. Measuring and interacting with processes involving no human vision

iv. Narrow band light sensors where [LEDs operate in a reverse-bias mode](https://en.wikipedia.org/wiki/LEDs_as_light_sensors) and respond to incident light, instead of emitting light

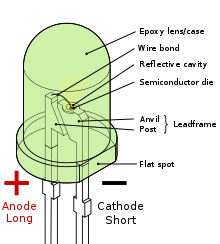
The [low energy consumption](https://en.wikipedia.org/wiki/Energy_conservation), low maintenance and small size of LEDs has led to uses as status indicators and displays on a variety of equipment and installations. Large-area [LED displays](https://en.wikipedia.org/wiki/LED_display) are used as stadium displays, dynamic decorative displays, and [dynamic message signs](https://en.wikipedia.org/wiki/Dynamic_message_sign) on freeways. Thin, lightweight message displays are used at airports and railway stations, and as [destination displays](https://en.wikipedia.org/wiki/Destination_sign) for trains, buses, trams, and ferries.

[](https://en.wikipedia.org/wiki/File:Red_and_green_traffic_signals,_Stamford_Road,_Singapore_-_20111210.jpg)

**Figure 2.7(a):** Red and green LED traffic signals[2]

One-color light is well suited for [traffic lights](https://en.wikipedia.org/wiki/Traffic_light) and signals, [exit signs](https://en.wikipedia.org/wiki/Exit_sign), [emergency vehicle lighting](https://en.wikipedia.org/wiki/Emergency_vehicle_lighting), ships' navigation lights, and [LED-based Christmas lights](https://en.wikipedia.org/wiki/Christmas_lighting_technology#LEDs)

Using LEDs has styling advantages because LEDs can form much thinner lights than incandescent lamps with [parabolic reflectors](https://en.wikipedia.org/wiki/Parabolic_reflector)[2].

 **Figure 2.7(b):** Internal Structure of a 2 Terminal LED[2]

## **2.8 Speaker**

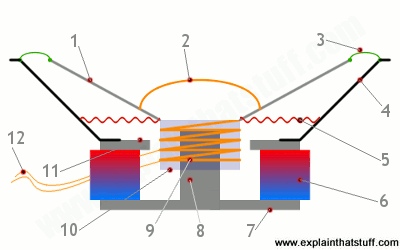
A speaker is used to translate an electrical signal into an audible sound, speakers contain an electromagnet: a metal coil which creates a magnetic field when an electric current flow through it. Inside a speaker, an electromagnet is placed in front of a permanent magnet.

At the front of a loudspeaker, there is a fabric, [plastic](https://www.explainthatstuff.com/plastics.html), [paper](https://www.explainthatstuff.com/papermaking.html), or lightweight metal cone (sometimes called a diaphragm) not unlike a drum skin (colored gray in our picture). The outer part of the cone is fastened to the outer part of the loudspeaker's circular metal rim, as seen in Figure 2.8(a). The inner part is fixed to an [iron](https://www.explainthatstuff.com/ironsteel.html) coil (sometimes called the voice coil, colored orange in the diagram) that sits just in front of a permanent [magnet](https://www.explainthatstuff.com/magnetism.html) (sometimes called the field magnet, and colored yellow). When you hook up the loudspeaker to a stereo, electrical signals feed through the speaker cables (red) into the coil. This turns the coil into a temporary magnet or electromagnet. As the [electricity](https://www.explainthatstuff.com/electricity.html) flows back and forth in the cables, the electromagnet either attracts or repels the permanent magnet. This moves the coil back and forward, pulling and pushing the loudspeaker cone. Like a drum skin vibrating back and forth, the moving cone pumps sounds out into the air[3].

All component parts can be seen in Figure 2.8(b).

**Figure 2.8(a):** Speaker

|  |
| --- |
|  |

 **Figure 2.8(b):** Internal structure of aspeaker[3].

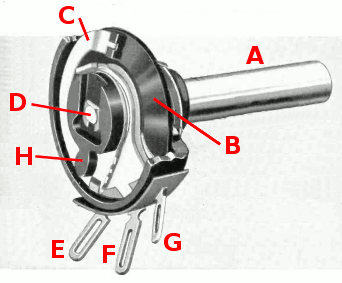
**Table 2.4:**  Labelling of **Figure 2.8(b)** parts and functions

|  |  |  |
| --- | --- | --- |
| **Part number** | **Name** | **Function and description** |
|  | Diaphragm (cone) | Moves in and out to push air and make sound. |
|  | Dust cap (dome) | Protects the voice coil from dust and dirt. |
|  | Surround | A piece of elastic [rubber](https://www.explainthatstuff.com/rubber.html), foam, or textile that flexibly fastens the diaphragm while allowing it to move freely |
|  | Basket | The sturdy metal framework around which the speaker is built. |
|  | Spider | A flexible, corrugated support that holds the voice coil in place, to the basket (outer frame) |
|  | Magnet | Typically made from ferrite or powerful neodymium. |
|  | Bottom plate | Made of soft iron. |
|  | Pole piece | Concentrates the magnetic field produced by the voice coil. |
|  | Voice oil | The coil that moves the diaphragm back and forth. |
|  | Former | A cylinder of cardboard or other material onto which the coil is wound. |
|  | Top plate | Also made of soft iron. |
|  | Cables | Connect stereo amplifier unit to voice coil. |
|  | | |

**2.9 Potentiometer**

A [potentiometer](https://en.wikipedia.org/wiki/Potentiometer) (colloquially, pot) is a three-terminal resistor with a continuously adjustable tapping point controlled by rotation of a shaft or knob or by a linear slider, as shown in Figure 2.9. The name potentiometer comes from its function as an adjustable [voltage divider](https://en.wikipedia.org/wiki/Voltage_divider) to provide a variable [potential](https://en.wikipedia.org/wiki/Potential) at the terminal connected to the tapping point. Volume control in an audio device is a common application of a potentiometer. A typical low power potentiometer (see drawing) is constructed of a flat resistance element of carbon composition, metal film, or conductive plastic, with a springy [phosphor bronze](https://en.wikipedia.org/wiki/Phosphor_bronze) wiper contact which moves along the surface. An alternate construction is resistance wire wound on a form, with the wiper sliding axially along the coil. These have lower resolution, since as the wiper moves the resistance changes in steps equal to the resistance of a single turn.

High-resolution multiturn potentiometers are used in precision applications. These have wire-wound resistance elements typically wound on a helical mandrel, with the wiper moving on a helical track as the control is turned, making continuous contact with the wire. Some include a conductive-plastic resistance coating over the wire to improve resolution. These typically offer ten turns of their shafts to cover their full range. They are usually set with dials that include a simple turns counter and a graduated dial and can typically achieve three-digit resolution. Electronic analog computers used them in quantity for setting coefficients and delayed-sweep oscilloscopes of recent decades included one on their panels[1].



**Figure 2.9:** Diagram of potentiometer with case cut away[1]

**Table 2.5:** Labelling of **Figure 2.9**

|  |  |
| --- | --- |
| **Part Label** | **Name and description** |
| A | Shaft |
| B | stationary carbon composition resistance element |
| C | phosphor bronze wiper |
| D | shaft attached to wiper |
| E,F,G | terminals connected to ends of resistance element, (*F*) terminal connected to wiper. |

**2.10 Transistor**

A transistor is a [semiconductor device](https://en.wikipedia.org/wiki/Semiconductor_device) used to [amplify](https://en.wikipedia.org/wiki/Electronic_amplifier) or [switch](https://en.wikipedia.org/wiki/Switch) [electronic](https://en.wikipedia.org/wiki/Electronics) signals and [electrical power](https://en.wikipedia.org/wiki/Electrical_power). It is composed of [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) material usually with at least three [terminals](https://en.wikipedia.org/wiki/Terminal_(electronics)) for connection to an external circuit. A [voltage](https://en.wikipedia.org/wiki/Voltage) or [current](https://en.wikipedia.org/wiki/Electric_current) applied to one pair of the transistor's terminals controls the current through another pair of terminals. Because the controlled (output) [power](https://en.wikipedia.org/wiki/Electric_power) can be higher than the controlling (input) power, a transistor can amplify a signal. Today, some transistors are packaged individually as seen in Figure 2.10(a), but many more are found embedded in [integrated circuits](https://en.wikipedia.org/wiki/Integrated_circuit).

Most transistors are made from very pure [silicon](https://en.wikipedia.org/wiki/Silicon), and some from [germanium](https://en.wikipedia.org/wiki/Germanium), but certain other semiconductor materials can also be used. A transistor may have only one kind of charge carrier, in a field-effect transistor, or may have two kinds of charge carriers in [bipolar junction transistor](https://en.wikipedia.org/wiki/Bipolar_junction_transistor) devices. Compared with the [vacuum tube](https://en.wikipedia.org/wiki/Vacuum_tube), transistors are generally smaller, and require less power to operate. Certain vacuum tubes have advantages over transistors at very high operating frequencies or high operating voltages. Many types of transistors are made to standardized specifications by multiple manufacturers[9].

** Figure 2.10(a):** Various kinds of transistors[9].

Transistors can either be used a switches or amplifiers.

**2.10(a) Transistor as a Switch**

Transistors are commonly used in [digital circuits](https://en.wikipedia.org/wiki/Digital_circuits) as electronic switches which can be either in an "on" or "off" state, both for high-power applications such as [switched-mode power supplies](https://en.wikipedia.org/wiki/Switched-mode_power_supply) and for low-power applications such as [logic gates](https://en.wikipedia.org/wiki/Logic_gate). Important parameters for this application include the current switched, the voltage handled, and the switching speed, characterised by the [rise and fall times](https://en.wikipedia.org/wiki/Rise_time).

In a grounded-emitter transistor circuit, such as the light-switch circuit shown, as the base voltage rises, the emitter and collector currents rise exponentially. The collector voltage drops because of reduced resistance from collector to emitter. If the voltage difference between the collector and emitter were zero (or near zero), the collector current would be limited only by the load resistance (light bulb) and the supply voltage. This is called saturation because current is flowing from collector to emitter freely. When saturated, the switch is said to be on [9].

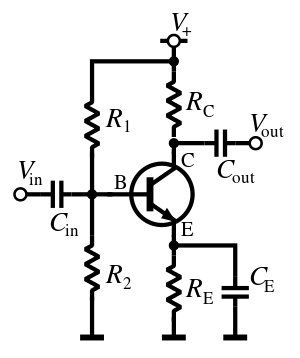
**2.10(b) Transistor as an Amplifier**

The [common-emitter amplifier](https://en.wikipedia.org/wiki/Common-emitter_amplifier), shown in Figure 2.10(b), is designed so that a small change in voltage (Vin) changes the small current through the base of the transistor; the transistor's current amplification combined with the properties of the circuit means that small swings in Vin produce large changes in Vout.

Various configurations of single transistor amplifier are possible, with some providing current gain, some voltage gain, and some both.

From [mobile phones](https://en.wikipedia.org/wiki/Mobile_phone) to [televisions](https://en.wikipedia.org/wiki/Television), vast numbers of products include amplifiers for [sound reproduction](https://en.wikipedia.org/wiki/Sound_reproduction), [radio transmission](https://en.wikipedia.org/wiki/Transmitter), and [signal processing](https://en.wikipedia.org/wiki/Signal_processing). The first discrete-transistor audio amplifiers barely supplied a few hundred milliwatts, but power and audio fidelity gradually increased as better transistors became available and amplifier architecture evolved.

Modern transistor audio amplifiers of up to a few hundred [watts](https://en.wikipedia.org/wiki/Watt) are common and relatively inexpensive[9].

[](https://en.wikipedia.org/wiki/File:NPN_common_emitter_AC.svg)

**Figure 2.10 (b**): Amplifier circuit, common-emitter configuration with a voltage-divider bias circuit[9].

**CHAPTER 3**

# **METHODOLOGY**

To achieve our objective of optimal security, it was important to utilize a method that will enable us to implement a voice, fingerprint and pin-requesting security access control system that has an audio and LED lighting output that indicates access granted.

The Yellow LED is always on to signify that the system is powered.

In this system, the PIN is the first level of security followed by the fingerprint.

Once an input PIN or fingerprint is correct, the green LED blinks

## **3.1 PIN Security Stage**

This pin security comprises of two different phases:

1. Registration phase
2. Recognition phase

In registration phase, the user is registered by inputting the pin on the keypad for the first time. This correct pin is stored on the Arduino Mega Memory. The pin can be changed hourly or daily depending on the level of the security required by the user.

The recognition phase involves the user inputting the pin on request. “ENTER PASSWORD” is shown on the LCD. If the right pin is inputted, the speaker gives an output “AFFIRMATIVE” which requires the user to proceed to the fingerprint stage and the LCD shows “PLACE FINGER” which moves the user to the next level of security. If the pin is wrong, the output audio is “NEGATIVE” and the LCD displays “ENTER PASSWORD” until the right pin is inputted.

## **3.2 Fingerprint Processing Stage**

Fingerprint processing is of two parts:

a. fingerprint enrollment

b. fingerprint matching (the matching can be 1:1 or 1: N).

Enrolling a new fingerprint requires changing the password as to update all the security levels. To enroll a fingerprint, the previous pin is required, then request for a new pin is shown then a prompt for enrolment of finger ID is shown.

When enrolling, the user enters the finger to capture their biometric data, then the system processes the finger biometric data to generate a template of the finger based on the previous processing results. This forms a part of the system database for future recognition.

For 1:1 matching, system will compare the live finger with specific template designed in the Module, for 1: N matching, or searching, system will search the whole finger library for the matching finger. In both circumstances, system will indicate a matching result, either a success or failure.

If successful, the audio output produced is “OPEN” the green LEDs come on to show access granted. If a failure occurs, the audio output is “FAILURE” the red LEDs come on to show access denied.

The LCD displays information to user at different stages during operation. It displays information to the user in response to the output given depending on if access is granted or denied. The speaker also gives an audio output for all states.

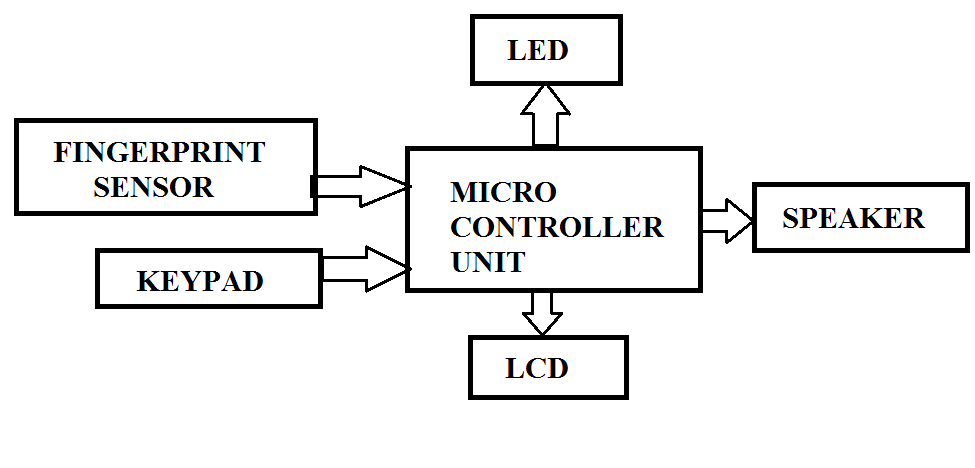
The recognition phase starts by requesting a password which when entered is compared with that stored in EEPROM of the system. If the entered password is the same as that stored in EEPROM, then the audio output of the system gives out a response of ‘AFFIRMATIVE’.

If the password entered is wrong, the output is ‘NEGATIVE’ and then we get an output to place finger and if finger placed can be matched to any fingerprint in memory of the fingerprint sensor then we get another ‘AFFIRMATIVE’ and ‘OPEN’ output, else the output will be ‘NEGATIVE’ and you will be asked to place your finger again.

**3.3 Block Diagram**

A block diagram shows the interaction between components of a system.

In Figure 3.1, the micro controller unit is the center of the whole system. The micro controller receives data from the Adafruit fingerprint sensor and keypad. The micro controller also sends out information to the LCD, LED and speaker

**Figure3.1**: Structure of hybrid access control system (Block Diagram)

## **3.4 Bubble Diagram**

A bubble diagram is a diagrammatic representation of a system showing the various states, inputs, outputs and transitions from state to state. A bubble diagram gives the total summary of a system, showing if it is time invariant or not. Also, it shows the hybrid nature of a system by considering if it is time automated. A bubble diagram also shows if a system is deterministic or not. A non-deterministic system is one in which one input leads to 2 different states.

The states of the system are usually represented in circles or ellipses, the transition is represented by an arc and the input and output are clearly defined on the transition arcs respectively.

The initial represented by an arc pointing to only that specific state.

The access control system can be modelled as a hybrid state machine with 4 states, as in Figure3.2:

a. System locked/active

b. PIN unlocked

c. Fingerprint Unlocked

d. System Unlocked

Clearly, the initial state of the system is the system locked/active state. Also, the access control system modelled is a time automated, time-invariant hybrid system. The various outputs produced from the input does not depend on the time the system is in use, hence time-invariant.

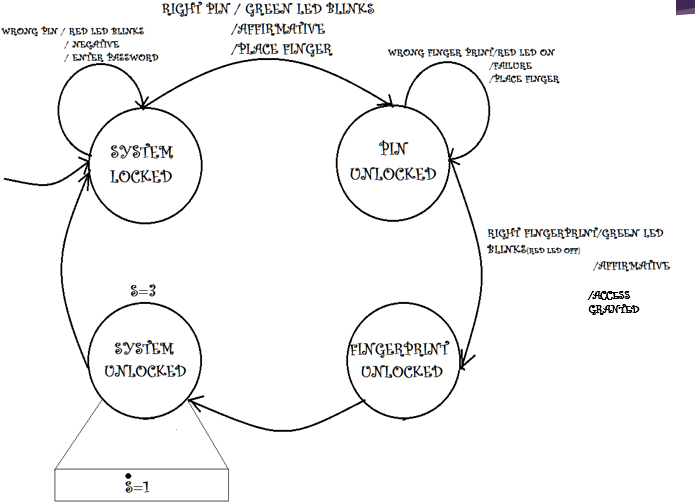
This system counts down from 3 to 1 which shows that it is time automated. Once the countdown is 3 seconds at the system unlocked state, a transition is made, and the system automatically locks, taking it back to the initial state

The inputs of the system are right PIN, wrong PIN, right fingerprint, wrong fingerprint.

Since the modelled access control system produces 3 outputs, instead of representing each in a separate bubble diagrams, a single bubble diagram is used with each output listed in a specific order. The order of output is LED output, audio output, LCD output. The various input-output relationship can be seen in Table3.1.

**Table 3.1**: Input-Output relationship of the bubble diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** | | | |
|  | **LED** | | **Audio** | **LCD** |
| Right PIN | Green Blinks | | Affirmative | Place Finger |
| Wrong PIN | Red On | | Negative | Input Password |
| Right Biometric | Green Blinks | | Open | Access Granted |
| Wrong Biometric | Red On | Failure | | Place Finger |



**Figure 3.2:** Bubble Diagram Showing the different states, output and input of the hybrid access control system.

# **CHAPTER 4**

# **RESULTS AND DISCUSSION**

**4.1 Results**

The circuit diagram in Figure 4.1 was simulated with Proteus 8. Figure 4.1 shows the ideal connection of all component parts.

The internal circuitry as seen in Figure 4.2 shows the complete connections of component parts and the interfacing between the two Arduino.

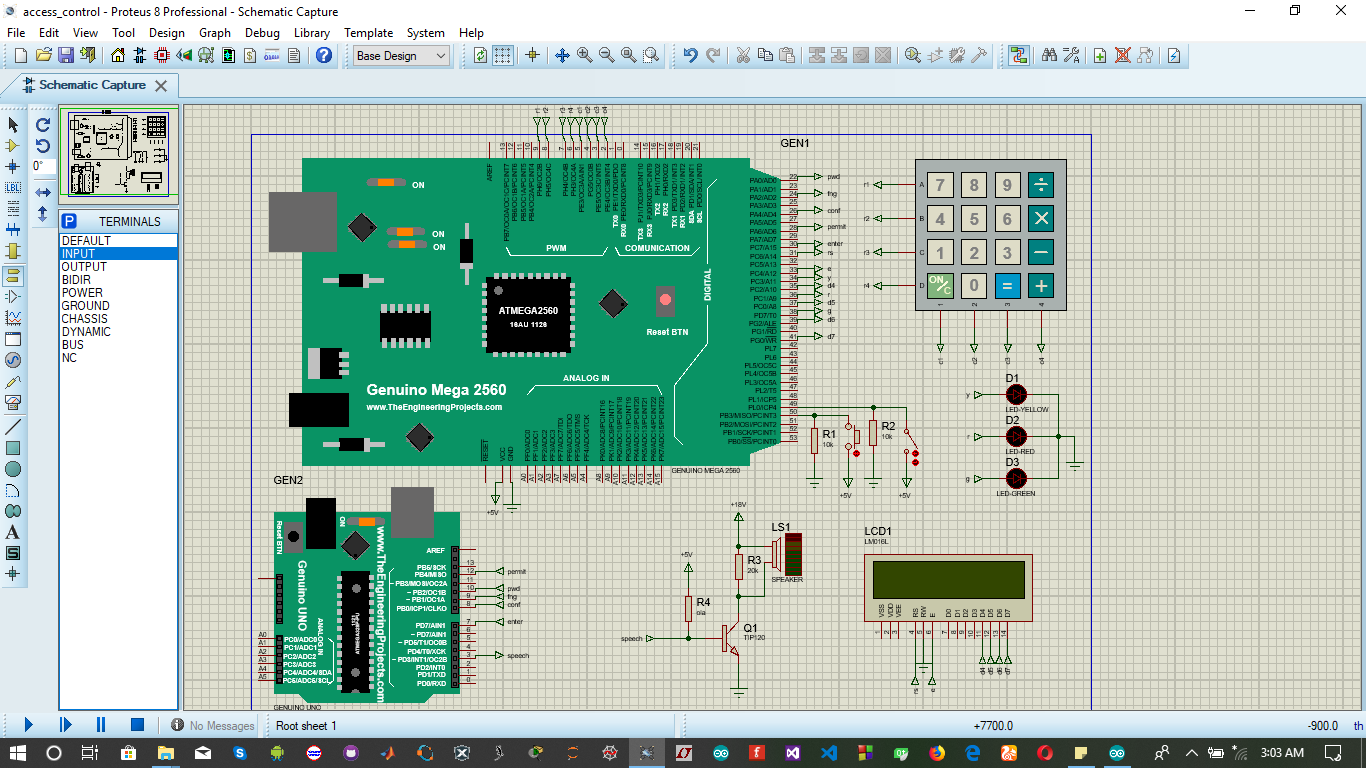
The access control hardware, as shown in Figure 4.3, has the LCD, LED and Fingerprint sensor externally placed. The speaker is embedded for portability.

The 2 Arduinos were programmed using The UNO IDE. The algorithm is shown in Appendix A1 and Appendix A2

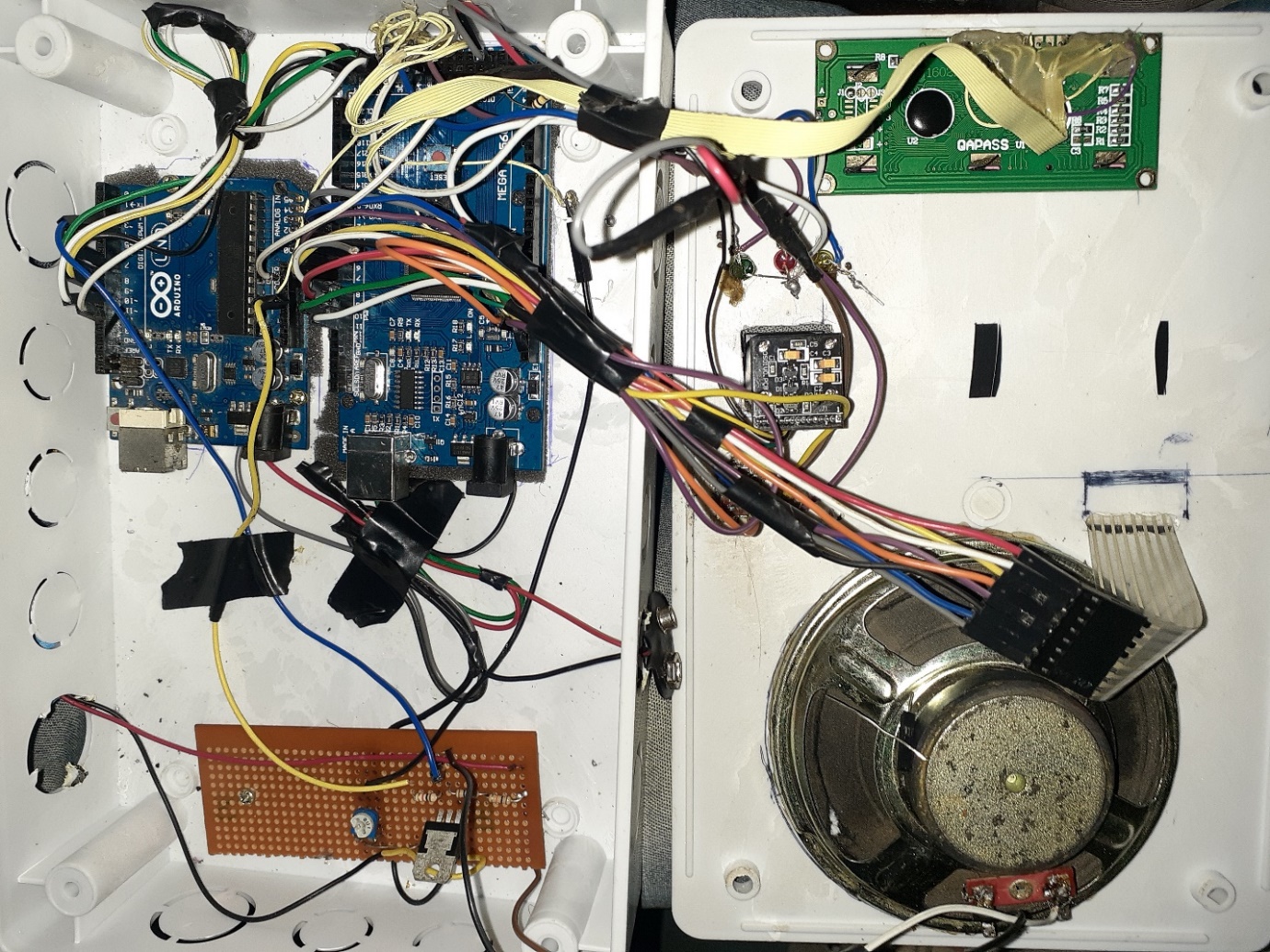
This system was able to differentiate between a wrong PIN input and a right one.

It was also able to differentiate successfully between the wrong / unregistered fingerprint and a right one

This system produced the required audio output with no noise because the voice was fully discretized. It also produced the required output on the LCD and LED indication.

****

**Figure 4.1:** Circuit Diagram of the Access Control System



**Figure4.2:** Internal Circuitry of Access Control System



**Figure 4.3:** External Hardware of Access Control System

## **4. 2 Limitations**

a. amplifier circuit is not so efficient, and this limits the audio output and so it is not loud enough.

1. Being a security system, it is meant to be powered always. Using a battery is also not an efficient system of power as the battery will end up being drained.
2. If one forgets the pin, the user cannot access the system.
3. This system cannot be used under certain weather conditions.
4. Talkie Library which is the audio library has very limited words.

## **CHAPTER 5**

**CONCLUSION AND FUTURE WORKS**

**5.1 Conclusion**

With this security system’s mode of operation, it can be inferred that the combination of pin lock and a biometric matching provides just enough security for versatile situations.

Attaching this system to an automated door can be done by attaching a motor and mechanical/magnetic lock mechanism.

Further modification such as adding a camera can allow for the system to be used in areas of open access to many users such as an Automated Teller Machine.

Also, modifications such as adding a storage device allows for the system’s history to be fully monitored.

This system can be modified efficiently by adding more levels of security.

## **Recommendations**

For increased efficiency, versatility and portability in advancements of future works on Access Control System, the following recommendations should be considered:

a. Future research should be done to improve the amplification of the audio output.

1. Further modifications such as adding a video camera to record the full activities of the system.
2. A storage system such as a computer should be attached to record the history of all activities. So, the authorized personnel can monitor the activities of the system thoroughly.
3. A more versatile audio library should be used. A text-to- speech module can also be included for optimum versatility.
4. A Wireless Local Area Network (WLAN) should be created to alert the administrator of the system at any point in time incase of intrusion.

**APPENDIX**

**A1. Algorithm**

#include <LiquidCrystal.h>

#include <EEPROM.h>

#include <Adafruit\_Fingerprint.h>

#include <SoftwareSerial.h>

#include <Key.h>

#include <Keypad.h>

//fingerprint init

LiquidCrystal lcd(31,33,35,37,39,41);

SoftwareSerial mySerial(10,11);

SoftwareSerial status(0,1);

Adafruit\_Fingerprint finger=Adafruit\_Fingerprint(&mySerial);

uint8\_t id;

//keyad init

const byte ROWS=4;

const byte COLUMNS=4;

char keys[COLUMNS][ROWS]={

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'\*','0','#','D'}

};

byte rowPins[ROWS]={9,8,7,6};

byte colPins[COLUMNS]={5,4,3,2};

Keypad customKeypad=Keypad(makeKeymap(keys),rowPins,colPins,ROWS,COLUMNS);

//access control parameters declaration

//inputs

int address=0;

bool not\_empty;

char df\_pwd[16];

char store\_pwd[16];

char ent\_pwd[16];

#define rst 50

#define del\_f 49

int kout=0;

//states

bool enrolled=false;

bool query\_pwd=false;

bool query\_finger=false;

bool reset=false;

bool check=true;

bool query=false;

bool achk=false;

bool finger\_enroll=false;

bool set\_pwd=false;

//outputs

int i,j=0;;

#define pwd 22

#define fng 24

#define conf 26

#define enter 28

#define permit 30

#define power 34

#define failed 36

#define success 38

void setup() {

// put your setup code here, to run once:

lcd.begin(16,2);

pinMode(power,OUTPUT);

pinMode(failed,OUTPUT);

pinMode(success,OUTPUT);

pinMode(rst, INPUT);

pinMode(del\_f, INPUT);

pinMode(pwd, OUTPUT);pinMode(fng, OUTPUT);pinMode(conf,OUTPUT);pinMode(enter, OUTPUT);pinMode(permit,OUTPUT);

digitalWrite(pwd, LOW);digitalWrite(fng, LOW),digitalWrite(conf,LOW);digitalWrite(enter,LOW);digitalWrite(permit,LOW);

Serial.begin(9600);

while(!Serial);

finger.begin(57600);

if(finger.verifyPassword()){

//Serial.println("fingerprint sensor found");

digitalWrite(power,HIGH);

}

else{

//Serial.println("fingerprint sensor not connected");

delay(1);

}

if(EEPROM.read(0)==(0))not\_empty=false;

else not\_empty=true;

if(not\_empty==false)first\_time();

else if(not\_empty==true)set\_password();

set\_password();

}

uint8\_t readnumber(void) {

uint8\_t num = 0;

while (num == 0) {

while (! Serial.available());

num = Serial.parseInt();

}

return num;

}

void loop() {

char key=customKeypad.getKey();

outwrite("pwd",0);outwrite("fng",0);outwrite("conf",0);outwrite("enter",0);

if((reset==false) and (finger\_enroll==false)){

lcd.setCursor(0,0);

lcd.print("enter password");

digitalWrite(pwd, LOW);digitalWrite(fng, LOW),digitalWrite(conf,LOW);digitalWrite(enter,LOW);digitalWrite(permit,LOW);

}

if(digitalRead(rst)==HIGH){

reset=true;

enrolled=false;

i=0;

outwrite("enter",1);

delay(1000);

outwrite("enter",0);

//Serial.println("enter old password:");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("old password");

}else if(key){

if(reset==true){

enrolled=false;

query\_pwd=false;

set\_pwd=false;

finger\_enroll=false;

//face\_enroll=false;

check=false;

query=false;

achk=false;

if((key=='#') or (i==15)){

i=0;

query\_pwd=true;

}else{

lcd.setCursor(i,1);

ent\_pwd[i]=key;

//Serial.print(key);

lcd.print(key);

i++;

}

}

if(query\_pwd==true){

if(strcmp(df\_pwd,ent\_pwd)==0){

outwrite("conf",1);outwrite("pwd",1);

delay(2000);

outwrite("pwd",0);outwrite("conf",0);

reset=false;

set\_pwd=true;

memset(df\_pwd,0,sizeof(df\_pwd));

//Serial.println("set new password:");

lcd.clear();lcd.setCursor(0,0);lcd.print("new password");

digitalWrite(failed,LOW);digitalWrite(success,HIGH);delay(1000);digitalWrite(success,LOW);

memset(ent\_pwd,0,sizeof(ent\_pwd));

key='A';

}else{

reset=true;

outwrite("conf",0);outwrite("pwd",1);

delay(1000);

outwrite("pwd",0);outwrite("conf",0);

//Serial.println("wrong password enter old password:");

digitalWrite(failed,HIGH);

lcd.clear();

lcd.setCursor(0,0);lcd.print("old password:");

}

}

if(set\_pwd==true){

query\_pwd=false;

if((key=='#') or (i==15)){

key=' ';

finger\_enroll=true;

set\_password();

i=0;

j=0;

memset(ent\_pwd,0,sizeof(ent\_pwd));

//Serial.println("new password is");

//Serial.println(df\_pwd);

lcd.clear();lcd.print("finger reg");

key=' ';

}else{

lcd.setCursor(j,1);

EEPROM.write(j,key);

//Serial.print(key);

lcd.print(key);

i++;

j++;

}

}

if(finger\_enroll==true){

set\_pwd=false;

while(key==' '){key=customKeypad.getKey();}

id=key;

if(id==0){

return;

}

if(digitalRead(del\_f)==HIGH){

fingerprintdelete(id);

}

else if(digitalRead(del\_f)==LOW){

//Serial.print("Enrolling finger ");

//Serial.print(id);

while(!fingerprintenroll()){

if(fingerprintenroll())break;

}

}

enrolled=true;

}else{};

if(enrolled==true){

check=true;

finger\_enroll=false;

outwrite("enter",1);

delay(1000);

outwrite("enter",0);

key='A';

}

if(check==true){

enrolled=false;

if((key=='#') or (kout==15)){

kout=0;

reset=false;

query\_pwd=false;

set\_pwd=false;

finger\_enroll=false;

achk=true;

query=true;

}else{

lcd.setCursor(kout,1);

query\_pwd=false;

ent\_pwd[kout]=key;

lcd.print(key);

//Serial.print(key);

kout++;

}

}

}

if(achk==true){

set\_password();

if(query==true){

//Serial.println();

if(strcmp(df\_pwd, ent\_pwd)==0){

digitalWrite(failed,LOW);digitalWrite(success,HIGH);delay(1000);digitalWrite(success,LOW);

outwrite("conf",1);outwrite("pwd",1);

delay(2000);

outwrite("pwd",0);outwrite("conf",0);

checkfingerprint();

query\_finger=false;

memset(ent\_pwd, 0,sizeof(ent\_pwd));

//code to read fingerprint

}else{

digitalWrite(failed,HIGH);

outwrite("conf",0);outwrite("pwd",1);

delay(2000);

outwrite("pwd",0);outwrite("conf",0);

outwrite("enter",1);

delay(1000);

outwrite("enter",0);

//Serial.print("wrong password");

memset(ent\_pwd, 0,sizeof(ent\_pwd));

lcd.clear();

}

}

}

achk=false;

}

int checkfingerprint(){

lcd.clear();lcd.setCursor(0,0);lcd.print("place finger");

//Serial.print("place finger");

int p=-1;

while(p!=FINGERPRINT\_OK){

p=finger.getImage();

switch(p){

case FINGERPRINT\_OK:

//Serial.println("fingerprint is being read");

break;

case FINGERPRINT\_IMAGEFAIL:

//Serial.println("please replace finger");

break;

}

p=finger.image2Tz();

while(p!=FINGERPRINT\_OK){

switch(p){

case FINGERPRINT\_OK:

//Serial.println("fingerprint read complete");

break;

case FINGERPRINT\_IMAGEMESS:

//Serial.println("fingerprint not clear, clean finger and wipe surface of fingerprint device");

break;

}

}

p=finger.fingerFastSearch();

if(p==FINGERPRINT\_OK){digitalWrite(failed,LOW);digitalWrite(success,HIGH);

outwrite("conf",1);outwrite("permit",1);

delay(1000);

outwrite("permit",0);outwrite("conf",0);

digitalWrite(success,LOW);

lcd.setCursor(0,1);lcd.print("access granted");

//Serial.println("access granted");

delay(2500);

lcd.clear();

}

else if(p==FINGERPRINT\_NOTFOUND){

digitalWrite(failed,HIGH);

outwrite("conf",0);outwrite("permit",1);

delay(1000);

outwrite("permit",0);outwrite("conf",0);

//Serial.println("fingerprint not found, replace finger, make sure to wipe finger and fingerprint device surface");

lcd.setCursor(0,1);lcd.print("access denied");

checkfingerprint();

return p;

}

}

}

uint8\_t fingerprintenroll() {

lcd.setCursor(0,0);lcd.print("enroll finger");

outwrite("fng",0);

int p = -1;

while (p != FINGERPRINT\_OK) {

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image taken");

break;

case FINGERPRINT\_NOFINGER:

//Serial.println(".");

break;

case FINGERPRINT\_PACKETRECIEVEERR:

//Serial.println("Communication error");

break;

case FINGERPRINT\_IMAGEFAIL:

//Serial.println("Imaging error");

break;

default:

//Serial.println("Unknown error");

break;

}

}

// OK success!

p = finger.image2Tz(1);

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image converted");

break;

case FINGERPRINT\_IMAGEMESS:

//Serial.println("Image too messy");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

//Serial.println("Communication error");

return p;

case FINGERPRINT\_FEATUREFAIL:

//Serial.println("Could not find fingerprint features");

return p;

case FINGERPRINT\_INVALIDIMAGE:

//Serial.println("Could not find fingerprint features");

return p;

default:

//Serial.println("Unknown error");

return p;

}

delay(2000);

p = 0;

while (p != FINGERPRINT\_NOFINGER) {

p = finger.getImage();

}

//Serial.print("ID "); Serial.println(id);

p = -1;

//Serial.println("Place same finger again");

lcd.setCursor(0,0);lcd.print("reenroll finger");

while (p != FINGERPRINT\_OK) {

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image taken");

break;

case FINGERPRINT\_NOFINGER:

//Serial.print(".");

break;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println("Communication error");

break;

case FINGERPRINT\_IMAGEFAIL:

//Serial.println("Imaging error");

break;

default:

//Serial.println("Unknown error");

break;

}

}

// OK success!

p = finger.image2Tz(2);

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image converted");

break;

case FINGERPRINT\_IMAGEMESS:

//Serial.println("Image too messy");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

//Serial.println("Communication error");

return p;

case FINGERPRINT\_FEATUREFAIL:

//Serial.println("Could not find fingerprint features");

return p;

case FINGERPRINT\_INVALIDIMAGE:

//Serial.println("Could not find fingerprint features");

return p;

default:

//Serial.println("Unknown error");

return p;

}

// OK converted!

//Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();

if (p == FINGERPRINT\_OK) {

//Serial.println("Prints matched!");

outwrite("conf",1);outwrite("fng",1);

delay(2000);

outwrite("fng",0);outwrite("conf",0);

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

//Serial.println("Communication error");

return p;

} else if (p == FINGERPRINT\_ENROLLMISMATCH) {

//Serial.println("Fingerprints did not match");

digitalWrite(failed,HIGH);

lcd.setCursor(0,1);lcd.print("failure");

outwrite("conf",0);outwrite("fng",1);

delay(1000);

outwrite("fng",0);outwrite("conf",0);

lcd.clear();

return p;

} else {

Serial.println("Unknown error");

return p;

}

Serial.print("ID "); Serial.println(id);

p = finger.storeModel(id);

if (p == FINGERPRINT\_OK) {

//Serial.println("Stored!");digitalWrite(failed,LOW);

digitalWrite(success,HIGH);

lcd.setCursor(0,1);lcd.print("enrolled");

delay(1500);

lcd.clear();

digitalWrite(success,LOW);

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

//Serial.println("Communication error");

outwrite("conf",0);outwrite("fng",1);

delay(1000);

outwrite("fng",0);outwrite("conf",0);

return p;

} else if (p == FINGERPRINT\_BADLOCATION) {

//Serial.println("Could not store in that location");

outwrite("conf",0);outwrite("fng",1);

delay(1000);

outwrite("fng",0);outwrite("conf",0);

return p;

} else if (p == FINGERPRINT\_FLASHERR) {

//Serial.println("Error writing to flash");

outwrite("fng",1);outwrite("conf",0);

delay(1000);

outwrite("conf",0);outwrite("fng",0);

return p;

} else {

//Serial.println("Unknown error");

outwrite("conf",0);outwrite("fng",1);

delay(1000);

outwrite("fng",0);outwrite("conf",0);

return p;

}

}

uint8\_t fingerprintdelete(uint8\_t id) {

uint8\_t p = -1;

lcd.clear();lcd.setCursor(0,0);lcd.print("delete finger");

p = finger.deleteModel(id);

if (p == FINGERPRINT\_OK) {

digitalWrite(failed,LOW);digitalWrite(success,HIGH);delay(1000);digitalWrite(success,LOW);

outwrite("conf",1);outwrite("fng",1);

delay(2000);

outwrite("fng",0);outwrite("conf",1);

//Serial.println("Deleted!");

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

//Serial.println("Communication error");

return p;

} else if (p == FINGERPRINT\_BADLOCATION) {

//Serial.println("Could not delete in that location");

return p;

} else if (p == FINGERPRINT\_FLASHERR) {

//Serial.println("Error writing to flash");

return p;

} else {

//Serial.print("Unknown error: 0x"); Serial.println(p, HEX);

return p;

}

}

void first\_time(){

String dft\_pwd="1234567";

unsigned int \_size=dft\_pwd.length();

i=0;

while(dft\_pwd[i]!='\0'){

EEPROM.write(address+i,dft\_pwd[i]);

i++;

}

EEPROM.write(address+\_size,'\0');

i=0;

}

void set\_password(){

i=0;

while(EEPROM.read(address+i)!='\0'){

df\_pwd[i]=EEPROM.read(address+i);

i++;

}

//df\_pwd[i++]='\0';

}

void clear\_eeprom(){

for (int i = 0 ; i < EEPROM.length() ; i++) {

EEPROM.write(i, 0);

}

}

void outwrite(String out,int val){

int in;

if(out=="pwd")in=22;

if(out=="fng")in=24;

if(out=="conf")in=26;

if(out=="enter")in=28;

if(out=="permit")in=30;

if(val==1)digitalWrite(in,HIGH);

else if(val==0)digitalWrite(in,LOW);

}

**A2. Audio algorithm**

#include <digitalWriteFast.h>

#include <Talkie.h>

#include <TalkieUtils.h>

#include <Vocab\_Soundbites.h>

#include <Vocab\_Special.h>

#include <Vocab\_Toms\_Diner.h>

#include <Vocab\_US\_Acorn.h>

#include <Vocab\_US\_Clock.h>

#include <Vocab\_US\_Large.h>

#include <Vocab\_US\_TI99.h>

Talkie ola;

#define permit 12

#define pwd 7

#define fng 10

#define conf 9

#define enter 8

void setup() {

pinMode(permit,INPUT);

pinMode(pwd,INPUT);

pinMode(fng,INPUT);

pinMode(conf,INPUT);

pinMode(enter,INPUT);

#define permits digitalRead(permit)

#define pwds digitalRead(pwd)

#define fngs digitalRead(fng)

#define confs digitalRead(conf)

#define enters digitalRead(enter)

}

void loop() {

if(enters==HIGH){

ola.say(sp2\_ENTER);

delay(900);

}

if((pwds==HIGH) and (confs==HIGH)){

ola.say(sp4\_AFFIRMATIVE);

delay(1900);

}

if((pwds==HIGH) and (confs==LOW)){

ola.say(sp4\_NEGATIVE);

delay(900);

}

if((fngs==HIGH) and (confs==HIGH)){

ola.say(sp4\_AFFIRMATIVE);

delay(1900);

}

if((fngs==HIGH) and (confs==LOW)){

ola.say(sp4\_FAILURE);

delay(900);

}

if((permits==HIGH) and (confs==HIGH)){

ola.say(sp4\_OPEN);

delay(900);

}

if((permits==HIGH) and (confs==LOW)){

ola.say(sp4\_SECURITY);

delay(900);

}

}

**A3. References**

[1] <https://en.wikipedia.org/wiki/Resistor#Potentiometers>

[2] <https://en.wikipedia.org/wiki/Light-emitting_diode>

[3] <https://www.explainthatstuff.com/loudspeakers.html>

[4] <https://simple.wikipedia.org/wiki/Fingerprint_scanner>

[5] <https://en.wikipedia.org/wiki/Arduino_Uno>

[6] <https://en.wikipedia.org/wiki/Electric_battery>

[7] [www.physics.org](http://www.physics.org)

[8] [www.explainthatstuff.com](http://www.explainthatstuff.com)

[9] <https://en.wikipedia.org/wiki/Transistor>

[10] <https://en.wikipedia.org/>