

DEDICATION

This project is dedicated to my parents for the moral and financial support and also to those who have guided me throughout my journey of education.

DECLARATION

I, OJO OLUWATOBILOBA JOHNSON do hereby declare that all the work in this project was carried out by me, under the supervision of ENGR.TEMIDAYO OFUSORI to the best of my knowledge, no such work has been submitted to the department of Electrical and Electronics Engineering, Federal University Oye-Ekiti.

Ojo Tobiloba Johnson

Date

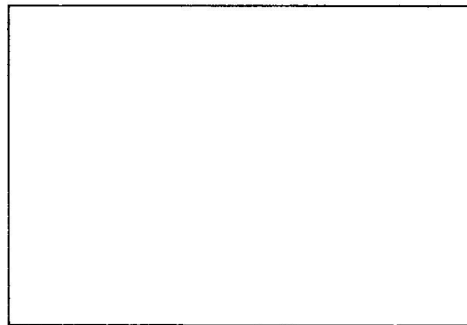
(EEE/13/1113)

Copyright

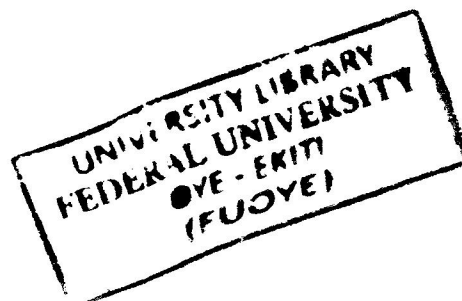
The copyright of this project and report belongs to Federal University, Oye-Ekiti.

Signed:

Date:



Office Stamp



CERTIFICATION

This is to certify that this project titled FINGERPRINT DOOR BASED AUTHENTICATION SYSTEM, by **OJO OLUWATOBILOBA JOHNSON** meets the minimum requirements governing the award of Bachelor Degree in Electrical and Electronics Engineering of Federal University Oye-Ekiti, Nigeria.

Ojo Tobiloba Johnson

(EEE/13/1113)

DATE

Engr. Temidayo Ofusori

(Project Supervisor)

DATE

Dr. (Engr.) Joseph Y. Oricha

(Head of Department)

DATE

ABSTRACT

Human identification field is very significant and has undergone rapid changes with time. An important and very reliable human identification method is fingerprint identification. Fingerprint of every person is unique, so this helps in identifying a person or in improving security of a system. This project uses a finger print sensor to read a user's identity in order to automatically operate a door. The finger print sensor is interfaced with a microcontroller (PIC16F877). The keypad attached is used to add, delete or identify a user by selecting the corresponding option on the keypad. The microcontroller enables the opening or closing of the door if there is a match between the scanned data and the already existing data. The comparison is done within the fingerprint module and the output sent to the micro-controller for appropriate action. The result of the output is also displayed on an LCD display which shows whether or not a user is authorized. The LCD also helps to make troubleshooting easier. The design also incorporates an alarm functionality to sensitize against unauthorized usage.

TABLE OF CONTENTS

Contents

DEDICATION	2
DECLARATION	3
CERTIFICATION	4
ABSTRACT.....	5
TABLE OF CONTENTS.....	6
LIST OF FIGURES	9
CHAPTER ONE	11
INTRODUCTION	11
1.1. PROJECT BACKGROUND:.....	11
1.2. PROBLEMS AND METHODOLOGY	11
1.3. AIM.....	12
1.4. OBJECTIVES	12
1.5. SCOPE OF PROJECT	12
CHAPTER TWO	14
THEORETICAL BACKGROUND.....	14
2.1. LITERATURE REVIEW	14
CHAPTER THREE	16
SYSTEM DESIGN	16
3.1. INTRODUCTION.....	16
3.2. POWER SUPPLY UNIT	16
3.2.1. SELECTION OF TRANSFORMER.....	17
3.2.2. SELECTION OF RECTIFIER	18
3.2.3. SELECTION OF FILTERING CAPACITOR	18
3.2.4. SELECTION OF VOLTAGE REGULATOR	19
3.3. MICROCONTROLLER	19
3.3.1. TYPES OF MICROCONTROLLER	19
3.3.1.1 Internal bus width	20
3.3.1.2. Instruction Set	20
3.3.1.3 Memory Architecture.....	21
3.3.1.4. Embedded and External Memory Architecture	21

3.3.2. The mid-range PIC microcontroller core features	21
3.3.3. Characteristics of PIC 16F877A.....	22
3.3.4. Core Architecture of PIC 16F877A.....	23
3.3.5. Data Space and Code Space of PIC 16F877A.....	24
3.3.6. Stacks of PIC 16F877A.....	24
3.3.7. Performance of PIC 16F877A	24
3.3.7.1. Single-word instruction.....	24
3.3.7.2. Instruction format.....	25
3.3.7.3. Input/output ports.....	25
3.3.7.4. Oscillator.....	26
3.3.7.5. Reset.....	27
3.3.7.6. Interrupts	28
3.3.7.7. Timers	28
3.3.7.8. Watchdog Timer	29
3.3.7.9. Sleep Mode	29
3.4. LIQUID CRYSTAL DISPLAY (LCD)	30
3.5. FINGERPRINT MODULE (FPM 10a).....	31
3.6. CRYSTAL OSCILLATOR.....	34
3.7. BUZZER	35
3.8. PROGRAMMING	35
3.8.1. Lower-level Programming.....	35
3.8.2. High level Programming	35
3.8.3. Machine code.....	35
CHAPTER FOUR.....	37
4.0. TESTING, ANALYSIS OF RESULTS AND DISCUSSION	37
4.1 TESTING	39
4.2 ANALYSIS.....	40
4.2.1 Classification and indexing of fingerprints	40
4.2.2. Fingerprint Features	40
4.2.3. Fingerprint Sensing	41
4.2.4. Feature Extraction Techniques.....	41
4.2.5. Fingerprint Matching Techniques	43

4.3. RESULTS.....	44
4.4 PROJECT MANAGEMENT	46
4.4.1. PROJECT SCHEDULE	46
4.5. RISK MANAGEMENT.....	47
4.6. SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL CONSIDERATIONS	47
CHAPTER FIVE	49
5.0. CONCLUSION AND RECOMMENDATION	49
5.1. CONTRIBUTION TO KNOWLEDGE.....	49
5.2. LIMITATIONS	50
5.3. FUTURE WORKS.....	50
5.4. CRITICAL APPRAISAL.....	50
BIBLIOGRAPHY AND REFERENCES	51
APPENDIX.....	52
APPENDIX I- GLOSSARY	52
APPENDIX II- TABLES.....	52
APPENDIX II-PROGRAM CODE	53

LIST OF FIGURES

Fig. 1: Block diagram of the Fingerprint door based authentication system.....	16
Fig. 2: Power supply stage	17
Fig. 3: Memory Architecture.....	21
Fig. 4: Block diagram of PIC 16F877A Architecture.....	23
Fig. 5: PIC16F877A.....	26
Fig. 6: Schematic Diagram of an LCD	31
Fig. 7: Fingerprint Module (FPM10A)	33
Fig. 8: Correct way to access Fingerprint sensor.....	34
Fig. 9: Crystal Oscillator.....	34
Fig. 10: Buzzer.....	35
Fig. 11: Circuit Diagram for the proposed project	37
Fig. 12: PCB circuit before etching.....	38
Fig. 13: After soldering	39
Fig. 14: Optical Sensor.....	41
Fig. 15: Fingerprint image labeled with different components.....	42
Fig. 16: Minutiae Extraction from Fingerprint Image.....	43
Fig. 17: Architecture of Fingerprint Matching Algorithm.....	43
Fig. 18: Fingerprint Enrollment.....	45
Fig. 19: Fingerprint Matching.....	45
Fig.20: Fingerprint Recognition (Authentication).....	46
Fig. 21: Gantt Chart.....	46
Fig. 22: Project Implementation Schedule on a Gantt Chart.....	47

ACKNOWLEDGMENTS

All glory, adoration, and gratitude to almighty God for sparing my life throughout this academic institution.

I greatly acknowledge the help of all the staff (Academic, Technologist, and non-Academics) in the department of Electrical and Electronics Engineering for their great impact in my life.

My sincere gratitude goes to my Supervisor Engr. Temidayo Ofusori for his advice, guidance and unflinching support throughout the duration of this project work.

My profound gratitude goes to the head of my department Engr. Dr. Joseph Oricha who has contributed immensely to my development both academically and morally.

It is also a pleasure to register my indebtedness to the contribution of my parents Mr. David .A Ojo and Mrs. Toyin Ojo who were behind my success, they supported morally, financially, academically, and spiritually just for me to be successful in life. Also, my profound gratitude goes to my staunch friends who really stood by me and supported me sincerely throughout my academic pursuit, in the person of Adesola Adebayo, John Samuel, and Engr. Ibrahim Adegoke.

Thanks to all that have in one way or the other contributed towards the successful completion of my course in the university, may God Almighty GOD bless you all.

To God be the glory.

CHAPTER ONE

INTRODUCTION

1.1. PROJECT BACKGROUND:

The fingerprint based authentication lock system is a reliable and very secure lock that will not only ensure safer environment but also ease lifestyle. This system can prove very useful in housing buildings, large offices, universities and so on. Because it offers the flexibility to add more features to the system. Users do not need to implement many systems from scratch. They can simply use the fingerprint lock system because fingerprint scanning is more accurate and cost effective method. It is also secure because fingerprint duplication is virtually impossible. Fingerprint is the preferred solution for these problems which provides high level of recognition accuracy. The skin on our palms and soles exhibits a flow like pattern of ridges called friction ridges. The pattern of friction ridges on each finger is unique and immutable. This makes fingerprint a unique identification for everyone. Fingerprint door lock incorporates the proven technology. Fingerprint scanner scans the finger-prints of users and its used for ensuring authentication. Fingerprint scanning is more accurate and cost effective method and duplication is virtually impossible. A Fingerprint recognition system can easily perform verification. In verification, the system compares an input fingerprint to the enrolled fingerprint of a specific user to determine if they are from the same finger. Now the security of our home/office is literally in our hands or rather on our fingertips.

1.2. PROBLEMS AND METHODOLOGY

PROBLEM STATEMENT

This project is motivated from few reasons which is listed below:

Carelessness by owners that hold the key

Owners have responsibility to take care of their own properties such as a house. The important things are the "KEYS". Without the key, owner or somebody else can't enter the house. So, if owner mistakenly misplace the keys then scenarios such as robbery or property lost are bound to occur.

The conventional key can easily be duplicated by thief.

Home security is the most important thing that needs to be given attention. Nowadays, every owner must install the security system that can be used to guard every home from thief. However, not all sees this to be important because they strongly believe the keys can secure the house properly from allowing any intruder from entering.

Restriction to some unauthorized buildings

There is an urgent need for the restriction to some buildings like Senate building, Board room conferences etc in the school from unauthorized students. This basically because of the fact that such buildings as listed above are known for discussing confidential matters unless they are granted the access of entry.

METHODOLOGY

The methodology in this project involves the software implementation and the Hardware implementation.

The software implementation includes:

- i. Definition of task
- ii. Designing the system
- iii. Writing the control program
- iv. Testing and debugging the program.

The Hardware implementation includes:

- i. Designing of the power supply circuit
- ii. Designing of the logical control unit
- iii. Designing of the microcontroller unit.
- iv. Designing of the display unit.
- v. integrating the power supply, logical control, microcontroller and the display units together.
- vi. Coupling of the whole design.

1.3. AIM

The aim of this project is to design and implement a finger print door based authentication system.

1.4. OBJECTIVES

The objectives of this project are:

- To design and develop a security system that use fingerprint biometrics.
- To develop a low cost finger print door based access control using finger print sensor.
- To add features to the system due to its flexibility.
- To eliminate the need for keeping track of keys or remembering a combination password, or PIN.

1.5. SCOPE OF PROJECT

The scope of work in this project is stated as given:

- Understanding microcontroller's structure and functionality.
- Understanding the PIC16F778A internal components specifically the EEPROM and timers and counters.

- Interfacing microcontrollers with input/output devices specifically the fingerprint, Liquid crystal display, buzzer etc.
- Developing a microcontroller program or code

CHAPTER TWO

THEORETICAL BACKGROUND

2.1. LITERATURE REVIEW

Biometrics based authentication applications that is critical to the growth of the global economy comprises of many features. These include but not limited to single sign-on, Web security, transaction security, application logon, data protections, workstations, remote access to resources (Maltoni, 2003). Most finger-scan technologies are based on minutiae. Samir Nanavati [3] states that 80 percent of finger-scan technologies are based on minutiae matching but that pattern matching is a leading alternative. This technology bases its feature extraction and template generation on a series of ridges, as opposed to discrete points. The use of multiple ridges reduces dependence on minutiae points, which tend to be affected by wear and tear. The downside of pattern matching is that it is more sensitive to the placement of the finger during verification and the created template is several times larger in byte size. Among the biometrics technologies, the fingerprint system is the most mature, widely used, acceptable, less costly and considered proof of evidence in the courts of laws all over the world (Jain et al., 2002). Fingerprints are commonly used because their configuration does not change throughout an individual's life time (Chaudhari et al., 2014; Jain et al., 2002). Even when fingerprints temporarily change slightly due to cuts and bruises, they tend to reappear after the healing process (Bhattacharya & Mali, 2013). This makes the fingerprints suitable for human identity verification and authentication (Jain et al., 2002 & Maltoni, Maio, Jain, Prabhakar, 2002).

Pavithra.B.C *etal* focused on the implementation of finger print based locker using microcontroller and MAX232. A step by step approach in designing the microcontroller based system for securing the transactions of the user and providing the security for the locker system and even more for the PASSPORT verification using a finger print scanner has been followed. The result obtained in providing the security is quite reliable in all the three modes. The system has successfully overcome some of the aspects existing with the present technologies, by the use of finger print Biometric as the authentication Technology. The stated project has being able to add a security feature which sensitize an unauthorized usage and also ease the of enrollment new users as there is no complex technical-know on how it operates.

Aditya Shankar *et. al* focused in this project on the replacement of conventional techniques of locking system. They replaced the old methods like lock & key system and password authentication system by the biometric system. They basically used fingerprints for the authentication system, the person whose fingerprint saved in the database can easily access the locker. They also provide an alarm system to alerting the neighbours if an unauthorized person or thief tries to access the locker. To prove that person authorized to open the locker door they need to scan their fingerprint images. The scanner is interfaced to 8051 microcontroller; this controller will be controlling the scanning process. They also provided a keypad for password after the fingerprint scanning. This two-step verification is for the double security. And a buzzer is provided for alarm in case of unauthorized access of locker. Microcontrollers generally come

with various type of capabilities, 8051 microcontroller used in this research papers has a low storage capacity compared to the PIC 16F877A which makes it less efficient, gives limitation to the numbers of authorized users in its application. Also, the microcontroller to be used in this proposed project it's a flash type of memory which allows it to be programmed and erased while coupled to the circuit.

Atar Nasrin *etal* focused on the design and implement a locker high security system based on fingerprint, password and GSM technology which can be organized in banks, protected offices and homes. In this system bank will collect the biometric data of each person for assigning the lockers only authentic person can be recovered money, documents from the locker. An improvement was made in this literature as Password, GSM module is added to send messages for security purposes. The project stated took care of the unnecessary and incessant messages when an unauthorized user makes an attempt, also password makes the project complex if forgotten by the already registered thereby giving no access to such user.

Pavithra.b.c *et. al* mainly focused his project on security. They used R303A as a scanner. This module has in-built ROM, DSP and RAM. The fingerprint module has a capacity of storage 100 user's fingerprint. This module operates in 2 modes they are Master mode and User mode. Master mode is used to register the fingerprints which will be stored in the ROM present on the scanner with a unique ID. They provided a unique identification number for the last step of verification, which provides three wrong attempts. This research projects has a limited storage capacity which makes it impossible to be used in places like schools etc. An improvement is made in this proposed project by using a fingerprint module (fpm 10a) with storage capacity of 250 authorized users.

Sagar S. Palsodkar *etal* proposed project was for Bank lockers security system using biometric and GSM. In our proposed system first the user will enrol his user name password and his mobile number .then the camera of pc will automatically on and capture the face store with face id then the person will put finger on finger print module finger print will be scanned and stored with finger id. In this way user enrolment process will be completed. Camera of pc might not give a 100percent verification of the user depending on the sensitivity of such camera to be used for verification, the issue of the types of camera used in registration and verifications sometimes becomes an hindrance to easy accessibility. This project has being able to take care of unnecessary, unsolicited, disturbing messages that will be sent to the mobile number prior to the fact that an unauthorized user try gain access

Anil K.Jain (2007) focused on biometric template security which is an important issue because, unlike passwords and tokens, compromised biometric templates cannot be revoked and reissued. Protecting the template is a challenging task due to intra-user variability in the acquired biometric traits. He present an overview of various biometric template protection schemes and discuss their advantages and limitations in terms of security, revocability, and impact on matching accuracy. Template protection was taken care of which made it a need for a fingerprint module (fpm10a)of high sensitivity, large storage capacity, good template library to be employed in the state project topic

CHAPTER THREE

SYSTEM DESIGN

3.1. INTRODUCTION

This project takes cognisance of all the users by ensuring that their fingerprints are registered before each can gain entry or have access to a door enclosed subject. This is basically made possible with various components like fingerprint module (fpm 10a) which is the main part of this project without which user cannot open his/her locker; the fingerprint used is an optical scanner. Microcontroller (PIC 16F877A) is interfaced with other components through different ports. PIC16F877A is connected to Buzzer, fingerprint module, LCD (Liquid Crystal Display), keypad. Here the microcontroller is to reduce the price of the door locker because the microcontroller is very commercial so that anybody can afford the locker, Buzzer, this is a sounder that gives alarm for every unregistered or unauthorized personnel from gaining entry into the door enclosed subject; this component makes the project be used for security reasons. Liquid Crystal Display (LCD) this shows up to give instructions on when to place fingers on the fingerprint module also shows the authentication of a personnel, Keypad is meant for the registration, deletion of fingerprints into its module. The presence of LEDs helps to give electrical signals to the door circuit if provided in the design as it dictates HIGH or LOW provided based on the authentication.

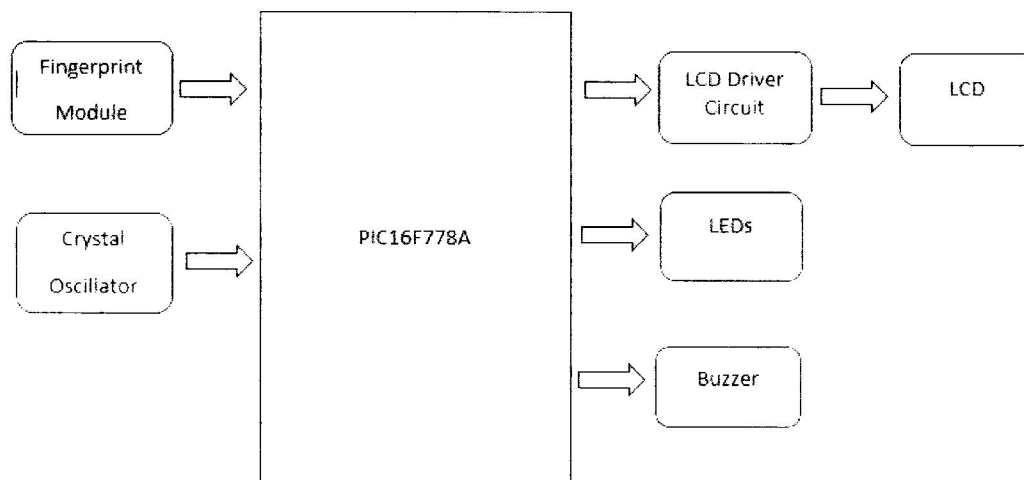


Fig 1: Block diagram of the Fingerprint door based authentication system

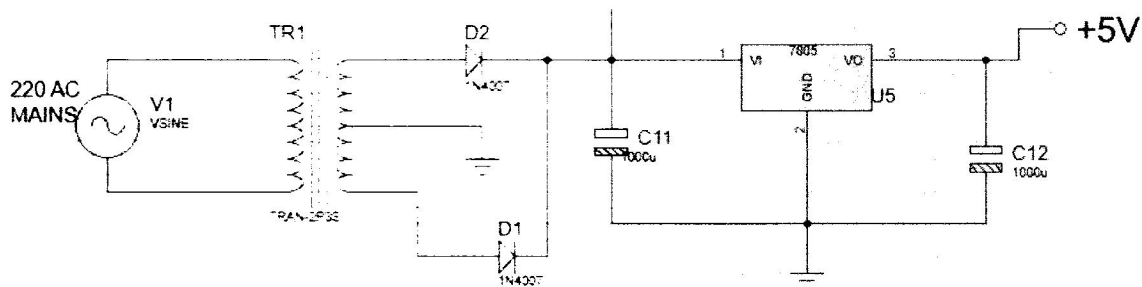
3.2. POWER SUPPLY UNIT

The power supply stage provides the appropriate DC voltage requirements to ensure the circuit components (especially the Active components) are powered properly. The circuit uses +5VDC supply. The power supply stage is a linear power supply type and involves a step down

transformer, Rectifier, Filter capacitor, and a voltage regulators, to give the regulated DC voltage. Power supplies in recent times have greatly improved in reliability but, because they have to handle considerably higher voltages and currents than any or most of the circuitry they supply, they are often the most susceptible to failure of any part of an electronic system. A DC Power Supply Unit (commonly called a PSU) deriving power from the AC mains (line) supply performs a number of tasks:

- It changes (in most cases reduces) the level of supply to a value suitable for driving the load circuit.
- It produces a DC supply from a pure AC wave.
- It prevents any AC from appearing at the supply output.
- It will ensure that the output voltage is kept at a constant level, independent of changes in:
 - a. The AC supply voltage at the supply input.
 - b. The Load current drawn from the supply output.
 - c. Temperature.

The circuit for the power supply stage is shown in figure 2.



3.2.1. SELECTION OF TRANSFORMER

In a basic power supply the input power transformer has its primary winding connected to the mains (line) supply. A secondary winding, electro-magnetically coupled but electrically isolated from the primary is used to obtain an AC voltage of suitable amplitude, and after further processing by the PSU, to drive the electronics circuit it is to supply. The transformer stage must be able to supply the current needed. If too small a transformer is used, it is likely that the power supply's ability to maintain full output voltage at full output current will be impaired. With too small a transformer, the losses will increase dramatically as full load is placed on the transformer. As the transformer is likely to be the most costly item in the power supply unit, careful consideration must be given to balancing cost with likely current requirement. In order to achieve the required DC power by the IC's, a step down transformer of primary windings voltage of 220V and secondary winding voltage of 12V was used. The transformer was assumed in an ideal situation which implies a unity power factor.

For an ideal transformer, we have;

$$\frac{V1}{V2} = \frac{N1}{N2}$$

Therefore, $\frac{220}{12} = \frac{55}{3}$

This implies that the turns ratio for the transformer used is 55:3

3.2.2. SELECTION OF RECTIFIER

If a transformer with a centre tapped secondary winding is used, more efficient full wave rectification can be used. The centre-tapped secondary produces two anti-phase outputs. If each of these outputs is 'half wave rectified' by one of the two diodes, with each diode conducting on alternate half cycles, two pulses of current occur at every cycle, instead of once per cycle in half wave rectification. The output frequency of the full wave rectifier is therefore twice the input frequency. This effectively provides twice the output voltage of the half wave circuit, $V_{pk} \times 0.637$ instead of $V_{pk} \times 0.318$ as the 'missing' half cycle is now rectified, reducing the power wasted in the half wave circuit. The higher output frequency also makes the smoothing of any remaining AC ripple easier. Although this full wave design is more efficient than the half wave, it requires a centre tapped (and therefore more expensive) transformer. IN4007 Diode was used in the rectifying circuit for it is perfect for most low voltage circuits where the current is less than 1A. It was ensured that the current flowing through these Diodes from the transformer output would not cause any damage since their forward voltage drop is almost constant irrespective of the current passing through them.

3.2.3. SELECTION OF FILTERING CAPACITOR

A typical power supply filter circuit can be best understood by dividing the circuit into two parts, the reservoir capacitor and the low pass filter. Each of these parts contributes to removing the remaining AC pulses, but in different ways. The capacitor that acts as temporary storage for the power supply output current. The rectifier diode supplies current to charge a reservoir capacitor on each cycle of the input wave. Of course, even though the reservoir capacitor has large value, it discharges as it supplies the load, and its voltage falls, but not by very much. At some point during the next cycle of the mains input, the rectifier input voltage rises above the voltage on the partly discharged capacitor and the reservoir is re-charged to the peak value V_{pk} again. With full wave rectification the performance of the reservoir capacitor in removing AC ripple is significantly better than with half wave, for the same size of reservoir capacitor, the ripple is about half the amplitude of that in half wave supplies, because in full

wave circuits, discharge periods are shorter with the reservoir capacitor being recharged at twice the frequency of the half wave design. Although a useable power supply can be made using only a reservoir capacitor to remove AC ripple, it is usually necessary to also include a low pass filter and/or a regulator stage after the reservoir capacitor to remove any remaining AC ripple and improve the stabilisation of the DC output voltage under variable load conditions.

3.2.4. SELECTION OF VOLTAGE REGULATOR

A voltage regulator is one of the most widely used electronic circuitry in any device. A regulated voltage (without fluctuations & noise levels) is very important for the smooth functioning of many digital electronic devices. An IC based voltage regulator can be classified in different ways. A common type of classification is 3 terminal voltage regulator and 5 or multi terminal voltage regulator. Another popular way of classifying IC voltage regulators is by identifying them as linear voltage regulator & switching voltage regulator. There is a third set of classification as 1) Fixed voltage regulators (positive & negative) 2) Adjustable voltage regulators (positive & negative) and finally 3) Switching regulators. In the third classification, fixed & adjustable regulators are basically versions of linear voltage regulators. A common case is with micro controllers where a smooth regulated input voltage must be supplied for the micro controller to function smoothly. This is the last stage in ripple elimination and setting the DC output to a fixed value. In order to derive a quality precision, LM7805 is used in performing the task of AC to DC step down. Its features includes: Maximum output current of 1.5A, fixed output voltage of 5V, thermal shutdown, etc [8]. We ensured that the input voltage to a voltage regulator is within the stated input voltage range as a result, the designed input voltage of LM7805 regulator.

3.3. MICROCONTROLLER

A microcontroller is a small computer furnished in a single integrated circuit that needs support chips. Its principal nature is self-sufficiency and low cost. It is not intended to be used as a computing device in the conventional sense; that is, a microcontroller is not designed to be a data processing machine, but rather an intelligent core for a specialized dedicated system. Microcontrollers are embedded in many control, monitoring, and processing systems. Some are general-purpose devices but most microcontrollers are used in specialized systems such as washing machines, telephones, microwave ovens, automobiles, and weapons. A microcontroller usually includes a central processor, input and output ports, memory for program and data storage, an internal clock, and one or more peripheral devices such as timers, counters, analogue to-digital converters, serial communication facilities, and watchdog circuits [3].

3.3.1. TYPES OF MICROCONTROLLER

Microcontrollers can be classified according to the following:

- Internal bus width
- Instruction set
- Memory architecture
- Embedded and external memory microcontroller
- Integrated circuits (IC) chip or very large scale integration (VLSI) core, very high speed integrated circuits (VHSIC), hardware description language (VHDL or Verilog) file
- Family

3.3.1.1 Internal bus width

They are classified to 8-bit, 16-bit and 32-bit microcontrollers.

In an 8-bit microcontroller it contains an 8-bit bus and the arithmetic logic unit (ALU) performs the arithmetic and logic operations on a byte at an instruction. Examples include PICx, Intel 8031 8051 and Motorola MC68HC11 families.

In 16-bit microcontroller, contains a 16-bit bus and the arithmetic logic unit (ALU) performs arithmetic and logic operations on the operand words of 16 bits at the instructions. These microcontrollers have a greater advantage than 8-bit microcontrollers in that they provide better performance and precision than their 8-bit counter parts. Typical examples include Intel 8096, PIC2x, Extended 8051XA, Atmega 16 and Motorola MC68HC12 families.

In 32-bit microcontrollers, the internal bus for the data transfer operations in this microcontroller is 32-bit. The ALU performs arithmetic and logic operations on operand words of 32 bits at the instructions. They give better performance and precision as compared to 16-bit microcontrollers but are more expensive. Some of the most popular 32-bit microcontrollers are PIC3x, Atmega 32, Intel/Atmel 251 family, Motorola M683xx and ARM 7, 9 or 11 processor based families. They find applications in embedded computing systems for applications-MPEG processing, mobile phones, MP3 audio systems, image processing based products and aerospace systems.

3.3.1.2. Instruction Set

This includes the following:

- Complex instruction set computer (CISC)
- Reduced instruction set computer (RISC)

A RISC machine contains few instructions not more than 35 and each instruction performs more elementary operations. Consequences of this are a smaller silicon area, faster execution, reduced program size with fewer accesses to main memory and less power dissipation. The PIC designers have followed the RISC route.

While on the other hand in a CISC (Complex Instruction Set Computer) microcontroller both data and instructions are 8-bits wide, with a minimum of 200 instructions. Data and code are on the same bus and cannot be fetched simultaneously.

3.3.1.3 Memory Architecture

This is how exchange of data between the control processing unit (CPU) and memory is carried out. Memory architecture includes the following:

- ❖ Princeton (Von-Neumann) memory architecture microcontroller
- ❖ Harvard memory architecture microcontroller

Microcontrollers using the von architecture have only one memory block and one data bus. As all data are exchanged by use of one data bus, thus the data bus can be overloaded in applications requiring high computational power hence making it very slow and inefficient. The CPU can either read an instruction or read/write data from/to the memory. Both cannot occur at the same time since the instructions and data use the same bus system [6]. Harvard architecture is a computer design in which data and instruction use different buses and storage areas. In other words, data and instructions are not located in the same memory area but in separate ones. In this type of machine, the processor can read and write instructions and data to and from memory at the same time and this result in a faster, more complex machine.

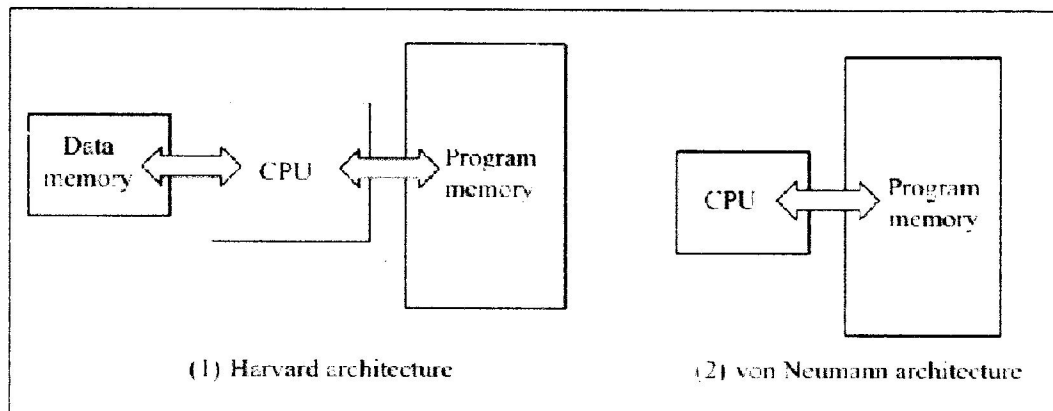


Figure 3: Memory Architecture

3.3.1.4. Embedded and External Memory Architecture

An embedded microcontroller has all the software and hardware units in a single integrated circuit while an external memory microcontroller does not have all the software and hardware units inside the integrated circuit. It may or may not have the entire memory unit externally interfaced. The interfacing circuit is known as a glue circuit.

3.3.2. The mid-range PIC microcontroller core features

PIC is a family of Harvard architecture microcontrollers made by Microchip technology, derived from the PIC1640, originally developed by the Microelectronics division of General Instrument. The name PIC initially is referred to Peripheral Interface Controller. PICs are popular with developers due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost, free development tools and serial programming (and reprogramming with flash memory) capability. The microcontroller is from PIC series. PIC microcontroller is the first reduced instruction set computing based microcontroller fabricated in complementary metal oxide semiconductor, that uses separate bus for instruction and data allowing simultaneous access of program and data memory. The main advantage of CMOS and RISC combination is low power consumption resulting in a very small chip size with a small pin count. The main advantage of CMOS is that, it has immunity to noise than other fabrication techniques.

Microcontrollers offer different kinds of memories. EEPROM, EPROM, FLASH etc. are some of the memories of which FLASH is the most recently developed. The technology used in PIC 16F877 is FLASH technology, so that data is retained even when the power is switched off. Easy programming and erasing are the other features of PIC 16F877. The heart of the microcontroller is the CPU core. In the past, this has traditionally been based on an 8-bit microprocessor unit. PIC micro devices are grouped by the size of the instruction word. The three current PIC micro families are:

Base-line: 12-bit instruction word length

Mid-range: 14-bit instruction word length

High-end: 16-bit instruction word length

PIC microcontrollers are unique in many ways. They have several general Characteristics which include: Harvard architecture, RISC processor design, single-word Instructions, machine and data memory configuration, and characteristic instruction formats.

3.3.3. Characteristics of PIC 16F877A

The PIC controller compared to other controllers is with low cost. The clock speed of the controllers is high with the rate of 20MHz. 8Kx14 words of FLASH program memory, 368x8 bytes of data memory (RAM), 256x8 bytes of EEPROM data memory and this is enough for the temperature control application. At the maximum clock rate, a PIC executes most of its instructions in 0.2 micro seconds or 5 instructions per microseconds. It has high speed in executing instruction. The efficiency and accuracy is very high. The instruction set consists of 35 instructions. For executing a program it requires only small steps. Power on reset and brown out protection ensure that the chip operates only when the supply voltage is within sections. A watch timer resets the PIC, if the chip malfunctions and deviates from its normal position. Any one of the core clock options can be supported including a low cost RC oscillator and a high accuracy crystal oscillator. These versatile timers can be characterized by inputs: control outputs and provide internal timing for program executions. The PIC microcontroller has a number of inbuilt

modules such as analog to digital converter, universal asynchronous transmitter and receiver that increases versatility of microcontroller. The PIC IC (Integrated Chip) is having wide operating voltage range from 2.5 to 6V. using power saving devices with a less power loss.

3.3.4. Core Architecture of PIC 16F877A

PIC controller architecture is simple. It is characterized by the features like using separate code and data spaces. The PIC controller is with small number of fixed length instructions and most of the instructions are single cycle execution (4 clock cycles), with single delay cycles upon branches and skips. The RAM location of PIC controller function as registers, as both source and destination of math and other functions. The data space mapped CPU, port, and peripheral registers in PIC 16F877A microcontroller and the program counter is mapped.

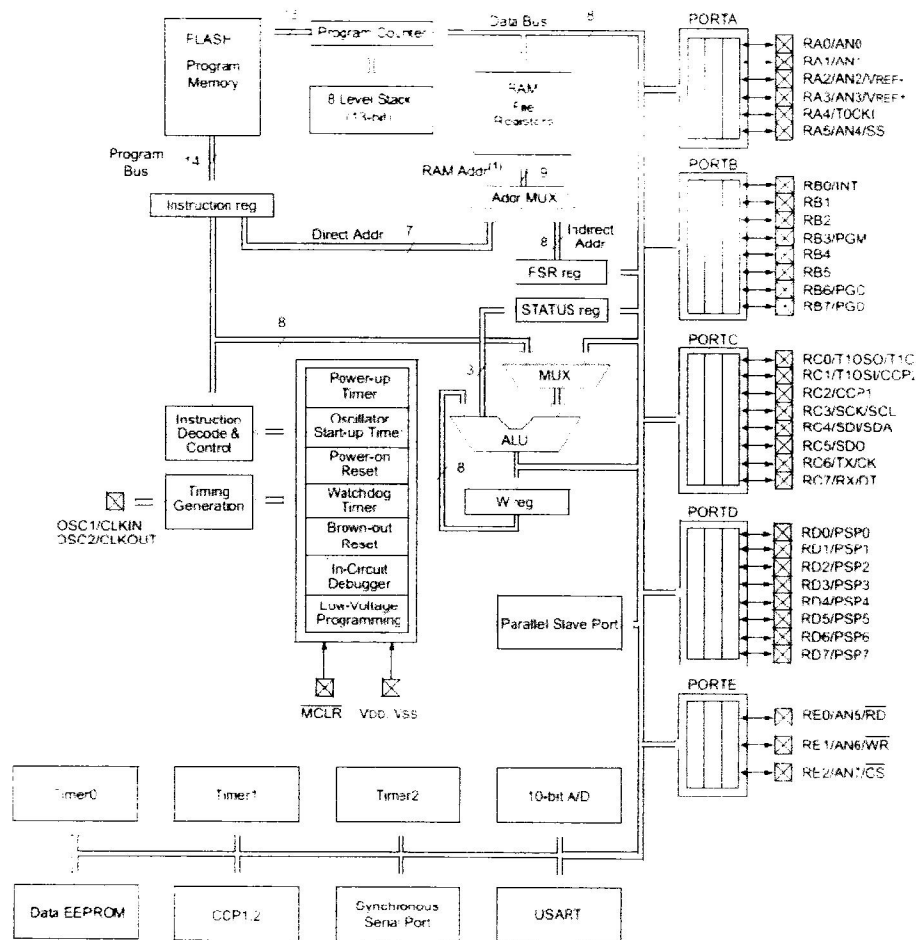


Figure 4: Block diagram of PIC 16F877A Architecture

3.3.5. Data Space and Code Space of PIC 16F877A

PICs have a set of registers that function as general purpose RAM. Special purpose control registers for on-chip hardware resources are also mapped into the data space. The addressability of memory varies depending on device series, and all PIC devices have some banking mechanism to extend the addressing to additional memory. In other microcontrollers, the register movement is achieved through the accumulator. External data memory is not directly addressable. PIC code space is generally implemented as EPROM, ROM, or FLASH ROM. In general, external code memory is not directly addressable, due to the lack of an external memory interface.

3.3.6. Stacks of PIC 16F877A

PICs have a hardware call stack, which is used to save return addresses. The hardware stack is not software accessible on earlier devices, but this is changed with the updated series devices. Hardware support for a general purpose parameter stack was lacking in early series, but this is greatly improved in the updated series, making the updated series architecture friendlier to high level language compilers.

3.3.7. Performance of PIC 16F877A

The clock speed is 20MHz. The TACQ or acquisition time is 19.2 μ s. The architectural decisions are directed at the maximization of top-end speed, or more precisely of speed to cost ratio. PIC architecture was among the first scalar CPU designs, and is still among the simplest and cheapest. The Harvard architecture in which instructions and data come conveniently from separate sources simplifies timing and microcircuit design greatly. PIC is particularly suited to the implementation of fast lookup tables in the program space. The optimization is facilitated by the relatively large program space of PIC and by the design of the instruction set, which allows for embedded constants. The simplicity of PIC, and its scalar nature, also serves to simplify significantly the construction of real time code. It is typically possible to multiply the line count of a PIC assembler listing by the instruction cycle time to determine execution time. The delay is constant even though instructions can take one or two instruction cycles, a dead cycle is optionally inserted into the interrupt response sequence to make the delay. External interrupts have to be synchronized with the four clock instruction cycle. Internal interrupts are already synchronized.

3.3.7.1. Single-word instruction

Since the device has separate buses for instructions and data, that is the Harvard architecture it is possible for instructions to be sized differently than data items. This ability to vary the number of bits in each instruction op-code makes it possible for the optimization

program memory and use of single-word instructions that can be fetched in one bus cycle. The PIC architecture has a two-stage instruction pipeline; however, since the fetch of the current instruction and the execution of the previous one can overlap in time, one complete instruction is fetched and executed at every machine cycle due to pipelining. The one exception is when an instruction modifies the contents of the Program Counter. In this case, a new instruction must be fetched, requiring an additional machine cycle. The PIC clocking system is designed so that an instruction is fetched, decoded, and executed every four clock cycles. In this manner, a PIC equipped with a 4MHz oscillator clock beats at a rate of 0.25 μ s. Since each instruction executes at every four clock cycles, hence each instruction takes 1 μ s

3.3.7.2. Instruction format

The mid-range family of PICs including the PIC16F877A have 14-bit instructions and a set of 35 instructions. The PIC16F877A instruction is a 14-bit word, divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The format for the instructions follows three different patterns:

- ❖ Byte-oriented operations
- ❖ Bit-oriented operations
- ❖ Literal and control operations

3.3.7.3. Input/output ports

Microcontrollers require to be connected to additional electronics, i.e. peripherals hence, microcontrollers have one or more registers (called "port" in this case) connected to the microcontroller pins, where the Port pins can be configured either as input or output, that is, general ports are bidirectional using specific special function registers. An important feature of I/O pins is the maximum current they can source/sink. For the most microcontrollers, current obtained from one pin is sufficient to activate low current devices (10- 20 mA). If a microcontroller has many I/O pins, then the maximum current of one pin is lower. Usually, each I/O port is under control by a special function register (SFR), this means that each bit of that register determines state of the corresponding pin. For example, by writing logic one (1) to one bit of that control register SFR, the appropriate port pin is automatically configured as input. It means that voltage brought to that pin can be read as logic 0 or 1. Otherwise, by writing zero to the SFR, the appropriate port pin is configured as output. Its voltage (0V or 5V) corresponds to the state of the appropriate bit of the port register. Another important pin feature is to (not) have pull-up resistors. These resistors connect pin to positive power supply voltage and their effect is visible when the pin is configured as input connected to mechanical switches or push buttons. Some versions of microcontrollers have pullup resistors connected to and disconnected from the pins by software.

PDIP

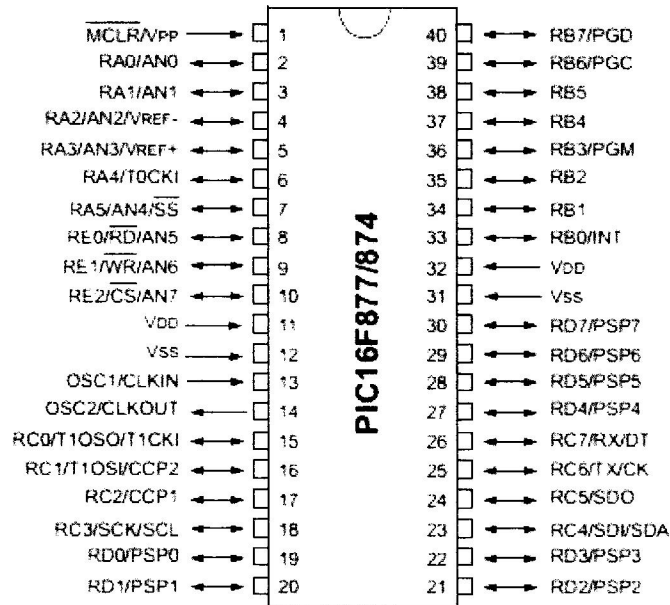


Figure 5: PIC 16F877A

3.3.7.4. Oscillator

This is a devices that produces a periodic waveform that is used as a synchronizing signal. Ideally the waveform are depicted as a square wave but they need not to be perfectly

symmetrical. The requirement of this waveforms is that they should be perfectly periodic. The basic timing interval for a digital circuit, which is equal to one full waveform period, is called the clock cycle. A stable and uniform waveform reaches exactly the same voltage every time the clock is high, and by the same token, every time the clock signal goes low the voltage level must be the same. The clock signal must remain at the high and low levels for the same time and the time between each high and low cycle must be exactly the same, this is the frequency stability of the clock. In practice, the stability and uniformity of the clock signal are more important than the absolute value. Another characteristic of the clock signal is the time required for clock levels to change from high to low and vice versa. Ideally this transition could be represented by a vertical line. This would mean that the transition is instantaneous, which is not achievable in actual circuits. Practically, some time is required for the waveform to transition from low to high and vice versa. Mid-range PICs use internal or external device to produce clock cycles which are required for their operation. The PIC executes an instruction every four clock cycles, so the oscillator speed determines the device performance. Mid-range PICs support up to eight different oscillator modes. The oscillator mode is selected at device programming time and cannot be changed at runtime. The configuration bits, which are non-volatile flags set during device programming, and this determine which oscillator mode is used by the program, among the following modes:

- ❖ Low Frequency Crystal (LP) 11
- ❖ Crystal Resonator (XT)
- ❖ High Speed Crystal Resonator (HS)
- ❖ External Resistor/Capacitor (RC)
- ❖ External Resistor/Capacitor (EXTRC)
- ❖ External Resistor/Capacitor (EXTRC) with clock out (CLKOUT)
- ❖ Internal 4 MHz Resistor/Capacitor (INTRC)
- ❖ Internal 4 MHz Resistor/Capacitor with CLKOUT (INTRC)

3.3.7.5. Reset

The reset mechanism places the PIC in a known condition. The reset mechanism is used to gain control of a runaway or hung-up program, as a forced interrupt in program execution, or to make the device ready at program load time. Practically this means that a microcontroller can behave rather inaccurately under certain undesirable conditions. In order to continue its proper functioning it has to be reset, meaning all registers would be placed in a starting position. Reset is not only used when microcontroller doesn't behave the way we want it to, but can also be used when trying out a device as an interrupt in program execution. The mid-range PICs are capable of several reset actions:

- ❖ Reset during power on (POR).
- ❖ Master clear (MCLR) reset during normal operation.
- ❖ Reset during SLEEP mode.

- ❖ Watchdog timer reset (WDT).
- ❖ Brown-out reset (BOR).
- ❖ Parity error reset.

The first two reset sources in the above list are the most common. POR reset serves to bring all PIC registers to an initial state, including the program counter register. The second source of reset action takes place when the MCLR line is intentionally brought down, usually by the action of a push-button reset switch. This switch is useful during program development since it provides a way of forcefully restarting execution.

3.3.7.6. Interrupts

An interrupt is an asynchronous signal calling for processor attention, originating from hardware or software. The interrupt mechanism is a way to avoid wasting processor time, since without interrupts code has to poll hardware devices using closed loops. With interrupts, the processor can continue to do its work since the interrupt mechanism ensures that the CPU receives a signal whenever an event occurs that requires its attention. After responding to or ignoring the interrupt, the CPU resumes processing where it left off. Processors and microprocessors usually support hardware and software interrupts and mask-able and non-mask-able interrupts; interrupts originate in practically any device connected to the system [3]. The interrupt source usually originates in one of the hardware modules, although some sources generate more than one interrupt.

3.3.7.7. Timers

These are used to carry out timing operations or counting operations simultaneously with the program, to make the program faster and more efficient [9]. Timer0 uses an 8-bit register, TMR0, file register. Its output is an overflow flag, timer0 interrupt flag (TOIF), bit 2 in the Interrupt Control Register (INTCON). The timer register is incremented via a clock input which is derived either from the microcontroller oscillator (FOSC) or an external pulse train at RA4. The register counts from 0 to 255 in binary, and then rolls over to 00 again. When the register goes from 0xFF to 0x00, TOIF is set. The timers are driven from the instruction clock, which can be monitored externally at the CLKOUT pin shown in figure 2.3, if the chip is operating with an RC clock. Alternatively, a count of external pulses can be made, and read from the register when finished, or the read triggered by external signal. Thus, the timers can also be used as counters. The pre-scaler is a divide by N register, where $N = 2, 4, 8, 16, 32, 64, 128$ or 256, meaning that the output count rate is reduced by this factor. This extends the count period or total count by the same ratio, giving a greater range to the measurement [9]. Equation below can be used;

$$T = \frac{C}{4PR}$$

Where T is the number of clock beats per second, C is the system clock speed in Hz, P is the value stored in the pre-scaler, and R is the number of iterations counted in the TMR0 register [3]. **Timer1** is a 16-bit counter, consisting of two 8-bit registers TMR1H and TMR1L. When the low byte rolls over from FF to 00, the high byte is incremented. The maximum count is therefore 65535, which allows a higher count without sacrificing accuracy.

Timer2 is an 8-bit counter (TMR2) with a 4-bit pre-scaler, 4-bit post-scaler and a comparator. It can be used to generate Pulse Width Modulated (PWM) output which is useful for driving DC motors and servos. These timers can also be used in capture and compare modes, which allow external signals to be more easily measured.

3.3.7.8. Watchdog Timer

The watchdog timer is designed to automatically reset the microcontroller if the program malfunctions, by stopping or getting stuck in loop. This could be caused by an undetected bug in the program, an unplanned sequence of inputs or supply fault. A separate internal oscillator and counter automatically generates a reset about every 18ms, unless this is disabled in the configuration word. If the watchdog timer is enabled, instructions which reset the watchdog timer are set on the appropriate program locations, besides commands which are regularly executed, then the operation of watchdog timer will not affect program execution. If for any reason (usually electrical noises in industry), the program counter “gets stuck” on some memory location from which there is no return, the watchdog will not be cleared and the register’s value being constantly incremented will reach the maximum hence Reset occurs.

3.3.7.9. Sleep Mode

The PIC microcontroller offer a built-in sleep modes where executing this instruction puts the microcontroller into a mode where the internal oscillator is stopped and the power consumption is reduced to an extremely low level. The main reason of using the sleep mode is to conserve the battery power when the microcontroller is not doing anything useful. The microcontroller usually wakes up from the sleep mode by external reset and interrupts or by a watchdog time-out.

3.4. LIQUID CRYSTAL DISPLAY (LCD)

This is a pixelated flat output device capable of displaying ASCII characters and dot-based graphics which can be monochrome or colour by use of the light modulating properties of liquid crystals which do not emit light directly. The LCD is very popular in the market due to its low cost and good contrast. It also has the distinct advantage over the light emitting diode (LED) in that it has a lower power consumption (typically of the order of microwatts as opposed to LEDs which is of order of milli-watts). It is due to low power consumption that there is compatibility with the metal oxide semiconductor (MOS) integrated circuit also LCDs have an incorporated controller for refreshing LCD, thereby relieving the CPU of the task of refreshing the LCD and also the ease of programming for characters and graphics. Nevertheless, the LCD suffers from a number of limitations including short operating life, poor visibility in low ambient lighting, requirement of an alternating current (AC) drive, limited temperature range (between 0 and 100 degree Celsius), slow speed and need of light source. The display used is a standard LM016L which displays 2 lines of 16 characters (16x2). Each character is 5x8 pixels, making it 80x16 pixels overall. The display receives ASCII codes for each character at the data inputs (D0–D7) when operating in 8bit mode but for the project it's operated in 4 bit mode hence the data inputs are D4 to D7. The data is presented to the display inputs by the PIC16F877A, and latched in by pulsing the E (Enable) input. The RW (Read/Write) is tied low (write mode), as the LCD is receiving data only. The RS (Register Select) input allows commands to be sent to the display. RS=0 selects command mode, RS=1 data mode. The display itself contains a microcontroller; the standard chip in this type of display is the Hitachi HD44780. It must be initialised according to the data and display options required. User interface Oscillator LCD PIC16F877A Buzzer Lock actuator 4x3 keypad 22 The LCD in the project is used in 4bit mode and hence requires only 6 port bits in the PIC16F877A instead of 10 port bits when used in 8bit mode, the LCD is used to display instructions, actions, options and data entered during operation of the digital lock. In the circuit diagram a variable resistor is present which is used to vary the brightness of the LCD. The LCD connection to the PIC16F877A. The LCD requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus). The three control lines are referred to as Enable (EN), Register Select (RS), and Read/Write (RW).

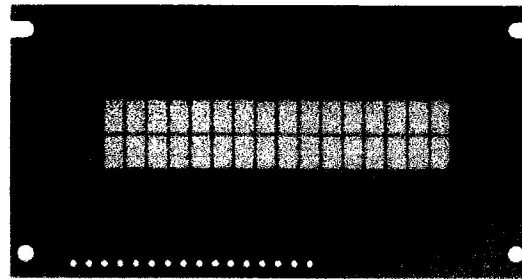
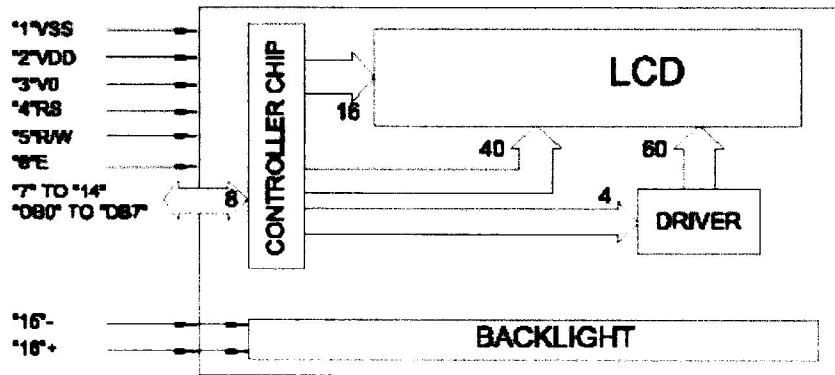


Figure 6: Schematic Diagram of an LCD Module

3.5. FINGERPRINT MODULE (FPM 10a)

The main part of this project is the fingerprint module without which user cannot open his/her locker. The fingerprint we used is an optical scanner, the heart of this scanner is a charged coupled device (CCD). The CCD has an array of light sensitive diodes, known as photosites. These photosites generate an electrical in response of light signals. These signals are stored in the form of dark and lights pixels for ridges and valleys respectively in the fingerprint module. these dark and light pixels are used to differentiate between different fingerprints. These scanned pixels collectively forms an image which is inverted. An analog-to-digital converter is present in the scanner which converts the analog electrical signals to the digital form (in the form of 0 and 1 which is a binary representation). Before comparing the scanned image of fingerprint to the prestored image, the scanner checks average darkness level of the pixels. it rejects if the scanned image is too dark or too light. Optical biometric fingerprint reader with great features

and can be embedded into a variety of end products, such as: access control, attendance, safety deposit box, car door locks. Fingerprint processing includes two parts; fingerprint enrollment and fingerprint matching. When enrolling, user needs to enter the finger two times. The system will process the two times Finger images, generate a template of the finger based on processing results and store the template. When matching, user enters the finger through optical sensor and system will generate a template of the finger and compare it with templates of the fingerprint library. For 1:1 matching, system will compare the live finger with specific templates designated 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 return the matching result success or failure.

Fingerprint is a pattern made up of ridges and valley on our fingertip skin. While storing the entry in database, scanner takes an image of these patterns and stores in its own memory. Then while performing search operation, it again takes pattern of fingerprint of that user who needs to gain access. This pattern is compared with all patterns previously stored in memory. In short it performs various iterations and executes matching algorithms and if it finds exact match then it gives out fingerprint ID number. Otherwise it gives out error signal.

Features:

- Integrated image collecting and algorithm chip together, All-in-one
- It can conduct secondary development, can be embedded into a variety of end products.
- Low power consumption, low cost, small size, excellent performance.
- Professional optical technology, precise module manufacturing techniques.
- Good image processing capabilities, can successfully capture image up to resolution 500dpi.

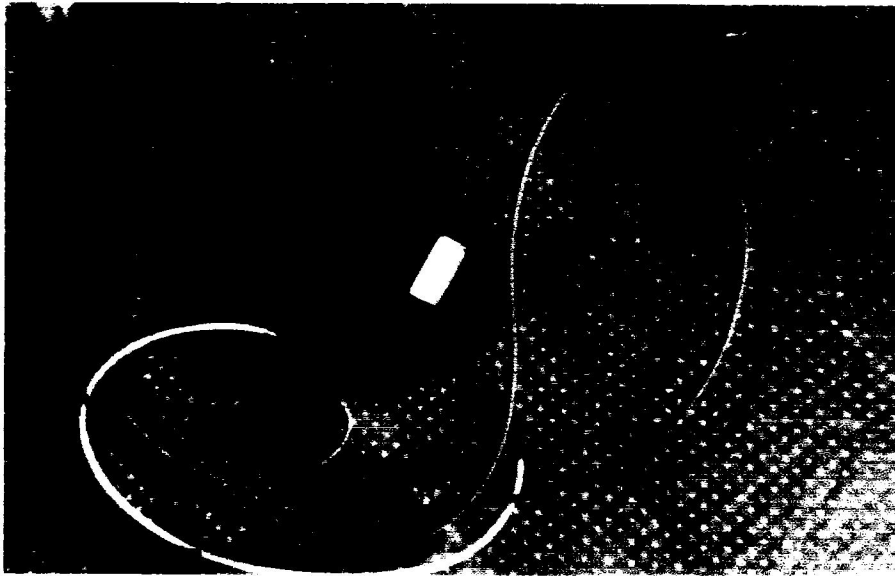
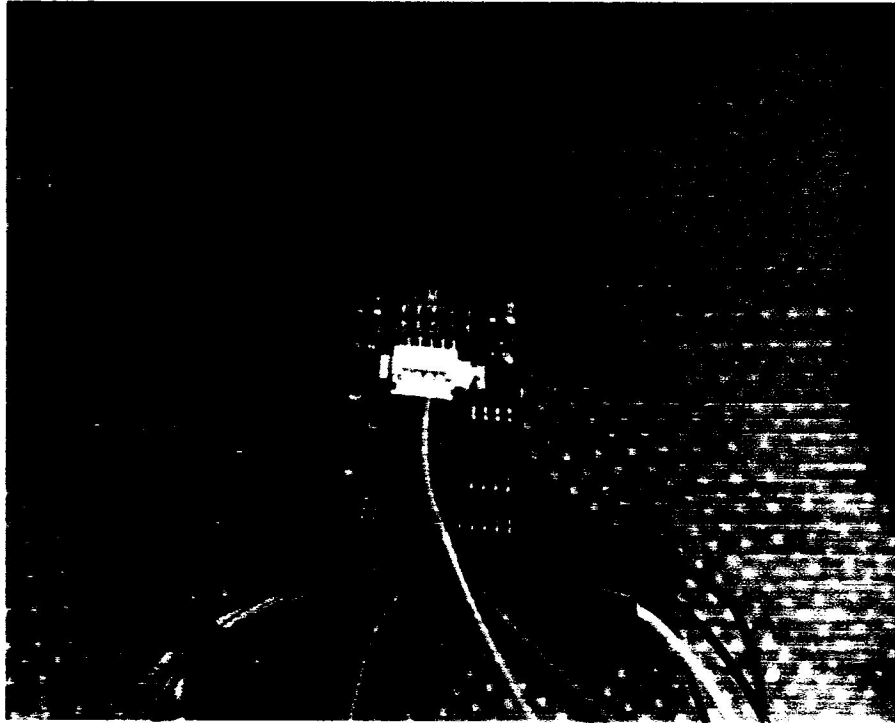


Fig 7: Fingerprint module (fpm10)

Precaution to be taken while accessing fingerprint sensor:

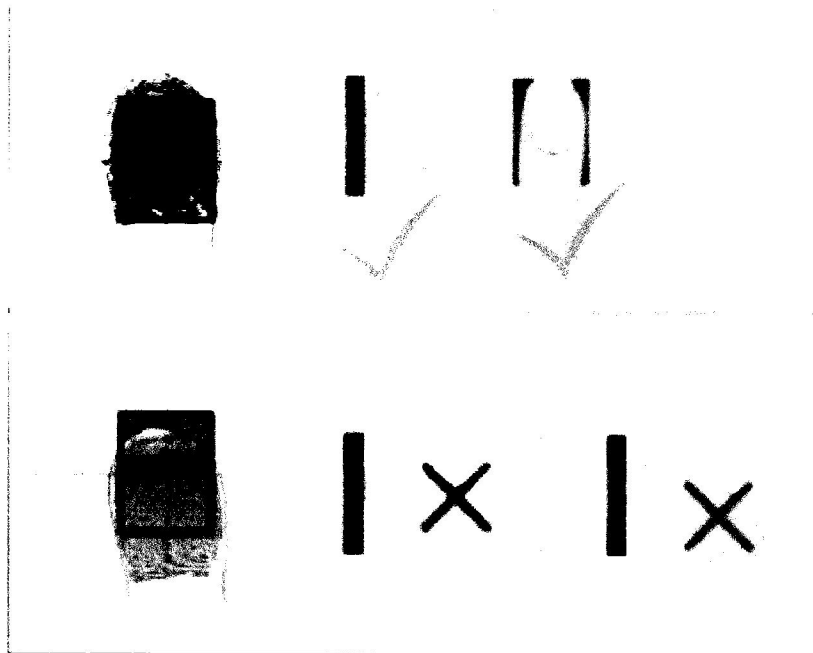


Figure 8: Correct way to access Fingerprint sensor.

3.6. CRYSTAL OSCILLATOR

They provide timing pulses for microprocessors. Crystal oscillators can be manufactured for oscillation over a wide range of frequencies, from a few kilohertz up to several hundred megahertz. Many applications call for a crystal oscillator frequency conveniently related to some other desired frequency, so hundreds of standard crystal frequencies are made in large quantities and stocked by electronics distributors. Using frequency dividers, frequency multipliers and phase locked loop circuits; it is practical to derive a wide range of frequencies from one reference frequency.

Every microcontroller has inbuilt oscillator circuit, connecting this quartz crystal with controller forms the crystal oscillator circuit which provide the desired frequency to the processor.



Fig. 9: Crystal Oscillator



3.7. BUZZER

A buzzer or beeper is an audio signalling device which may be mechanical, electro-mechanical or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Audio transducers and buzzers convert electrical energy into acoustic energy. These devices are found in smoke detectors and other products that produce sound.

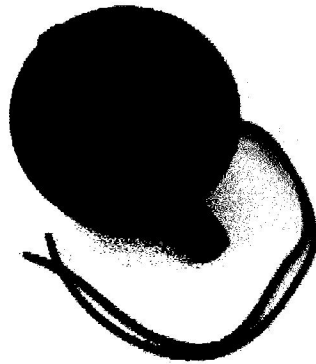


Fig 10: Buzzer

3.8. PROGRAMMING

This is the art of communication with devices using an artificial language that can be used to define a sequence of instructions that can be processed and executed by such devices. There are two broad classifications of programming languages – the low-level and the high-level languages.

3.8.1. Lower-level Programming

This type of language is written specifically for a particular type of micro-controller or microprocessor. This means that it cannot be used by another microcontroller. The instructions in this language are in mnemonics. This is called Assembly language.

3.8.2. High level Programming

This type of language is formed from parts of natural language such as English. It is a high level of abstraction between what is asked by the computer and what the computer actually understands. It is easily understood by humans more easily than assembly languages. But, like the assembly language the computer cannot understand it. They therefore have to be translated into machine code the language the computer understands.

3.8.3. Machine code

This is a sequence of carefully timed series of ON and OFF signals that can also be called high and low pulses or digital zeros and ones. The code usually represents numbers, data and instructions for manipulating those numbers and data.

The project was programmed in **ASSEMBLY LANGUAGE**. This program can be understood because it consists of alphabets signs and words. It is written in a program for text processing (editor) and is capable of producing an ASCII file on the computer disc or in specialized such as MPLAB.

CHAPTER FOUR

4.0. TESTING, ANALYSIS OF RESULTS AND DISCUSSION

Before the fabrication of this project work, **the circuit was simulated on Proteus Design Workbench** in order to have a prior, the behavioural pattern of the hardware when eventually deployed. Proteus is an application software that provides a virtual design environment for the testing of various electronic circuits.

Not all physical devices (such as sensors) can be realised in Proteus but these (unavailable devices) can be modelled in Proteus.

Below is the simulation snapshot of the project hardware (taking after the adopted circuit diagram) in Proteus:

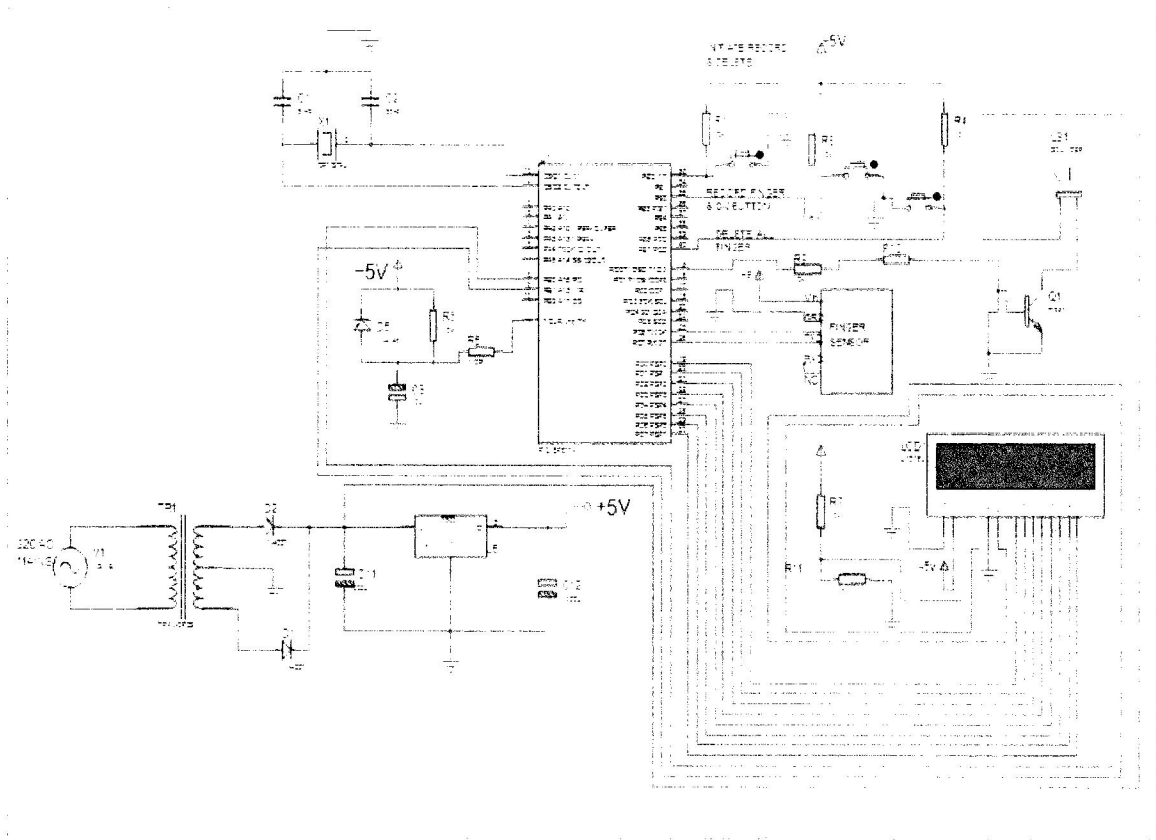


Figure 11: Circuit diagram for the proposed project

After the Successful Simulation of this circuit, the hardware fabrication was the next stage, as this was kick-started with the production of a PCB for the circuit. The production of PCB is done by etching the already designed circuit using the software Express PCB on a glossy paper to the vero-board using an etching chemical (FeCl_3). Then, sorting out was carried out on the PCB

circuit board so as to prevent any form of short circuit on the board, so as to prevent any form of damage. Images of the PCB layout and board are shown below respectively:

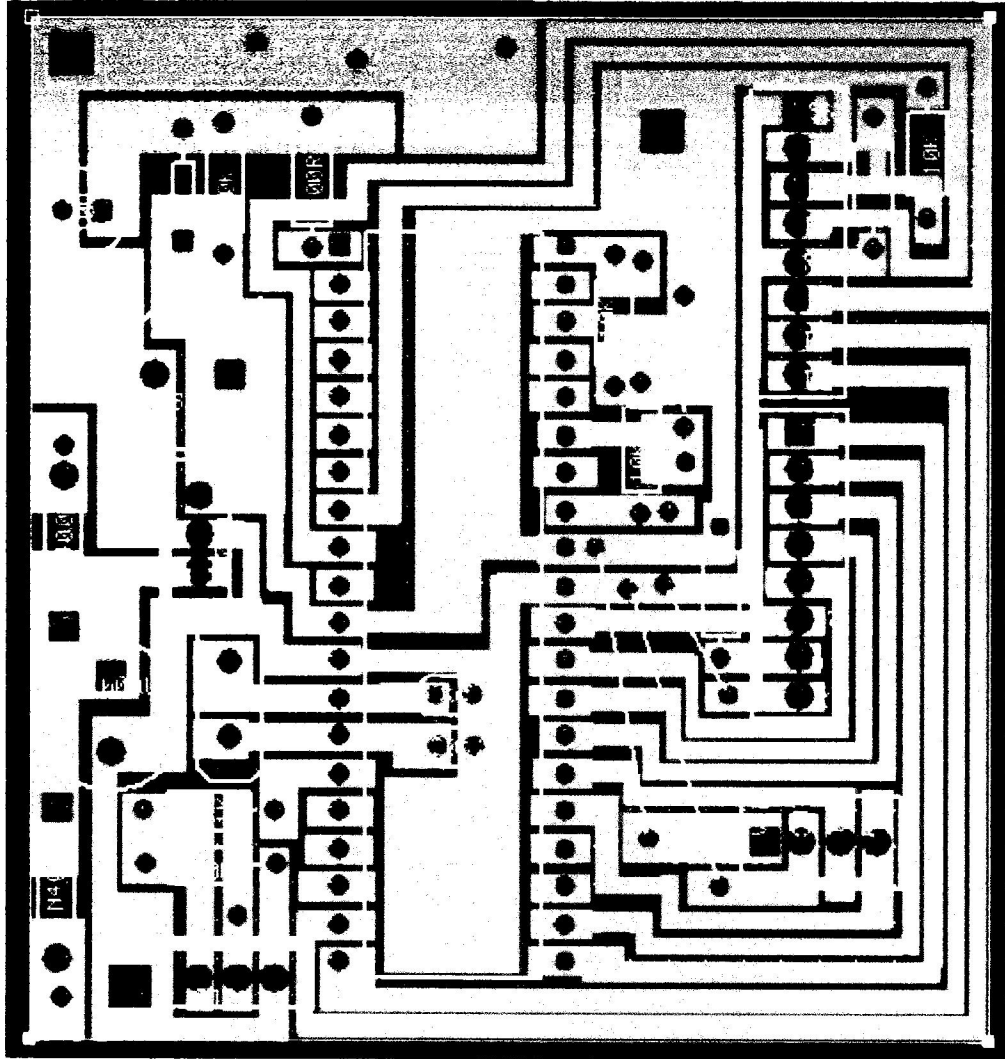


Fig 12: The PCB circuit before etching.

Soldering of all components in accordance to the adopted circuit diagram was done after the production of a PCB. The following precautions were taken during the soldering:

- i. While connecting the resistor to the PCB, the suitable spot was found and placed into its hole.
- ii. For a good heat transfer, the solder wire and the solder iron must be cleaned before starting.
- iii. In order to avoid the overheating of PCB, the components are usually elevated above the PCB.

- iv. After the component is inserted in the PCB hole, the excess lead is cut-off, thus leaving a length of about the radius of the pad.
- v. After soldering, the soldered joints must also be cleaned after it has been solidified.

During the fabrication process of the PCB, it was ascertained that the programming of the microcontroller has been done so as to provide room for correction while testing. Lastly, the implementation of the project was achieved and then packaged in a casing for mobility and ease.



Fig 13: After soldering

4.1 TESTING

Stage by stage testing was done according to the block representation on the breadboard, before soldering of circuit commenced on Vero board.

The process of testing and implementation involved the use of some test and measuring equipment stated below.

1. Bench Power Supply: This was used to supply voltage to the various stages of the circuit during the breadboard test before the power supply in the project was soldered. Also during the soldering of the project the power supply was still used to test various stages before they were finally soldered.

2. Oscilloscope: The oscilloscope was used to observe the ripples in the power supply waveform and to ensure that all waveforms were correct and their frequencies accurate. The waveform of the oscillation of the crystal oscillator used was monitor to ensure proper oscillation at 4MHz.
3. Digital Multi-meter: The digital multi-meter basically measures voltage, resistance, continuity, current, frequency, temperature and transistor βf . The process of implementation of the design on the board required the measurement of parameters like, voltage, continuity, current and resistance values of the components and in some cases frequency measurement. The digital multi-meter was used to check the output of the voltage regulators used in this project.

4.2 ANALYSIS

Individuality of fingerprints is based on empirical observations. However Golfarelli *et al* [6] formulated the optimum Bayesian decision criterion for a biometric verification system and obtained a theoretical equal error rate (EER) of 1.31×10^{-5} for a hand-geometry-based verification system and of 2×10^{-3} for a face-based verification system. Similarly there is limited probability of correspondence of two fingerprints.

4.2.1 Classification and indexing of fingerprints

Fingerprint authentication includes two subdomains: one is fingerprint verification (*Am I who I claim I am?*) and the other is fingerprint identification (*Who am I?*), the latter being more difficult requiring extensive indexing and classification of fingerprints for efficient retrieval. Nearly all fingerprint classification schemes used today are derived from the famous *Henry System* – a detailed fingerprint indexing method for aiding manual fingerprint comparison. For instance, the FBI uses one variant which recognizes eight different types of patterns: radial loop, ulnar loop, double loop, central pocket loop, plain arch, tented arch, plain whorl, and accidental. Whorls are usually circular or spiral in shape. Arches have a mound-like contour, while tented arches have a spikelike or steeple-like appearance in the center. Loops have concentric hairpin or staple-shaped ridges and are described as "radial" or "ulnar" to denote their slopes: ulnar loops slope toward the little finger side of the hand, radial loops toward the thumb. Fingerprint classification & indexing is a difficult pattern recognition problem due to small inter-class variability compared to large intra-class variations in fingerprint patterns.

4.2.2. Fingerprint Features

A fingerprint is an impression of the epidermal ridges of a human fingertip. A hierarchy of three levels of features, namely, Level 1 (pattern), Level 2 (minutiae points) and Level 3 (pores and ridge shape) are used for recognition purposes. Most AFISs employ Level 1 & Level 2 features. Level 1 features refer to the overall pattern shape of the unknown fingerprint—a whorl, loop or some other pattern. This level of detail cannot be used to individualize, but it can

help narrow down the search. Level 2 features refers to specific friction ridge paths — overall flow of the friction ridges and major ridge path deviations (ridge characteristics called minutiae) like ridge endings, lakes, islands, bifurcations, scars, incipient ridges, and flexion creases. Level 3 detail refers to the intrinsic detail present in a developed fingerprint — pores, ridge units, edge detail, scars etc. High resolution sensors (~1000dpi) are required for extraction of Level 3 features. But as EER values are reduced (relatively ~20%) using them along with Level 1 & 2 features. Moreover Level 3 features offer greater success in partial fingerprint recognition..

4.2.3. Fingerprint Sensing

Fingerprint sensing techniques can be of two types – *offline scanning* and *live-scanning*. In off-line sensing fingerprints are obtained on paper by “*ink technique*” which are then scanned using paper scanners to produce the digital image. Most AFISs use live-scanning where the prints are directly obtained using an electronic fingerprint scanner. Almost all the existing sensors belong to one of the three families: *optical*, *solid-state*, and *ultrasound*. Optical sensors, based on the frustrated total internal reflection (FTIR) technique are commonly used to capture live-scan fingerprints in forensic and government applications. They are the most common fingerprint sensors. An important breakthrough in sensor technology was the development of optical sensors based on fiber-optics leading to sensor miniaturization and enhanced portability.

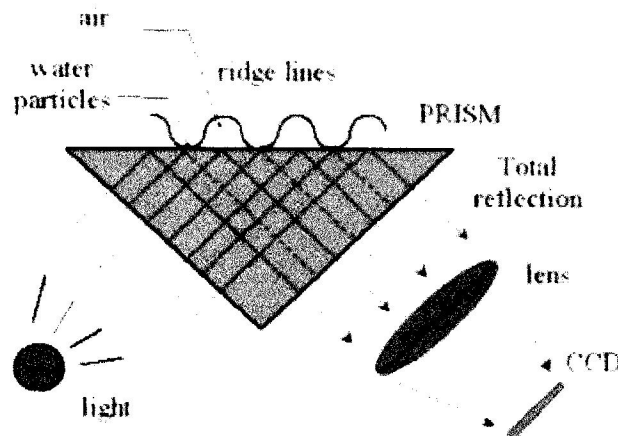


Fig 14: Optical Sensor

4.2.4. Feature Extraction Techniques

For the purpose of automation, a suitable representation i.e. feature extraction of fingerprints is essential. This representation should have the following properties –

- Retention of discriminating power of each fingerprint at several levels of resolution
- Easy computability
- Amenable to automated matching algorithms

- Stable and invariant to noise and distortions
- Efficient and compact representation

Several feature extraction methods have been proposed and implemented successfully over the years. Roughly speaking there are four categories of methods based on fingerprint feature extraction by image processing. The first category of methods extract minutiae directly from the gray-level image without using binarization and thinning processes while the second category extracts features from binary image profile patterns. The third category of methods uses machine learning for extracting minutiae and the last category extracts minutiae from binary skeletons.



Fig 15: A fingerprint image labeled with different components.

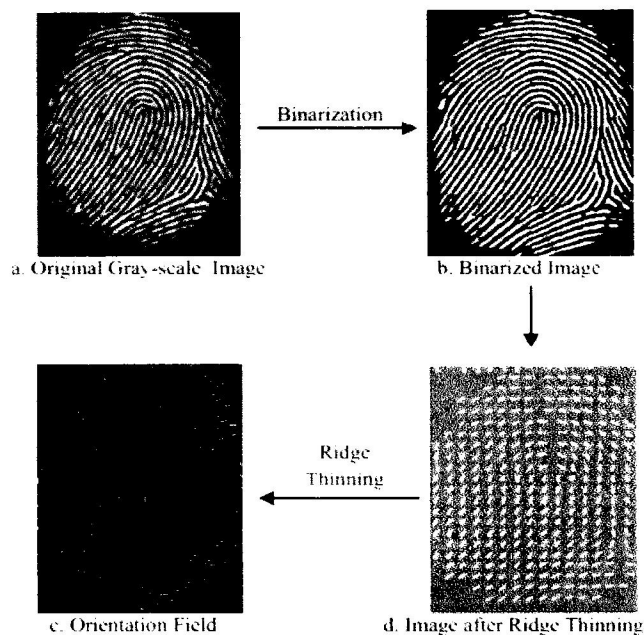


Fig 16: Minutiae Extraction from fingerprint image

Binarization is the process by which an enhanced grey level image is transformed into a binary image for subsequent feature detection. Good binarization algorithms should minimize information loss and also provide efficient computational complexity. A binarization approach based on the peak detection in the cross section gray-level profiles orthogonal to the local ridge orientation

4.2.5. Fingerprint Matching Techniques

Matching fingerprint images is an extremely difficult problem, mainly due to the large variability in different impressions of the same finger (i.e., large *intra-class* variations). Fingerprint matching algorithms are roughly classified into 3 major categories –

- i. *Correlation-based Matching* :
Two fingerprint images are superimposed and the correlation between corresponding pixels is computed for different alignments (e.g. various displacements and rotations). Fourier transform as well as Fourier-Mellin Transform can be used to speed up the correlation computation
- ii. *Feature-based (or Minutiae-based) Matching* :
Typical fingerprint recognition methods employ feature based matching, where minutiae (i.e., ridge ending and ridge bifurcation) are extracted from the registered fingerprint image and the input fingerprint image, and the number of corresponding minutiae pairings between the two images is used to recognize a valid fingerprint image.

iii. *Pattern-based (or Image-based) Matching:*

Pattern based algorithms compare the basic fingerprint patterns (e.g., local orientation and frequency, ridge shape, texture information) between a previously stored template and a candidate fingerprint. The images need to be aligned in the same position, about a central point on each image. The candidate fingerprint image is then graphically compared with the template to determine the degree of match.

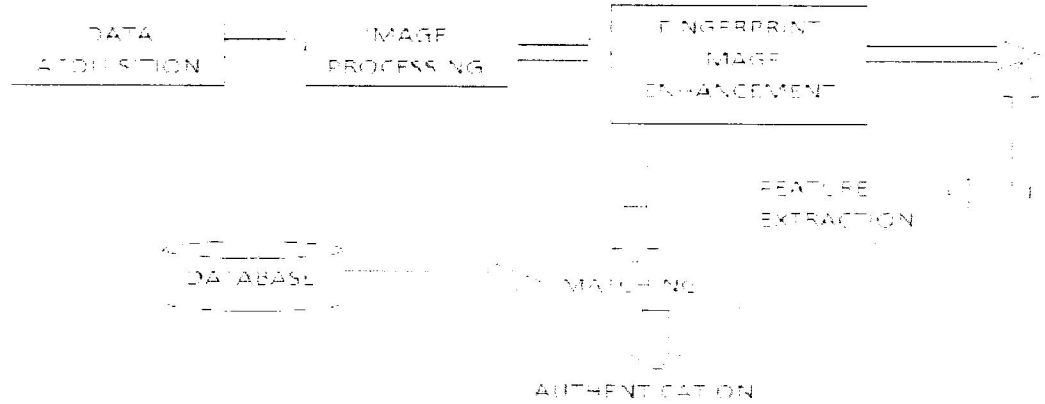


Fig 17: Architecture of fingerprint matching algorithm

4.3. RESULTS

The design includes the following process such as fingerprint enrollment, fingerprint matching with the database, Authentication. Each of these processes has been achieved in the design which is therefore shown below as the results:

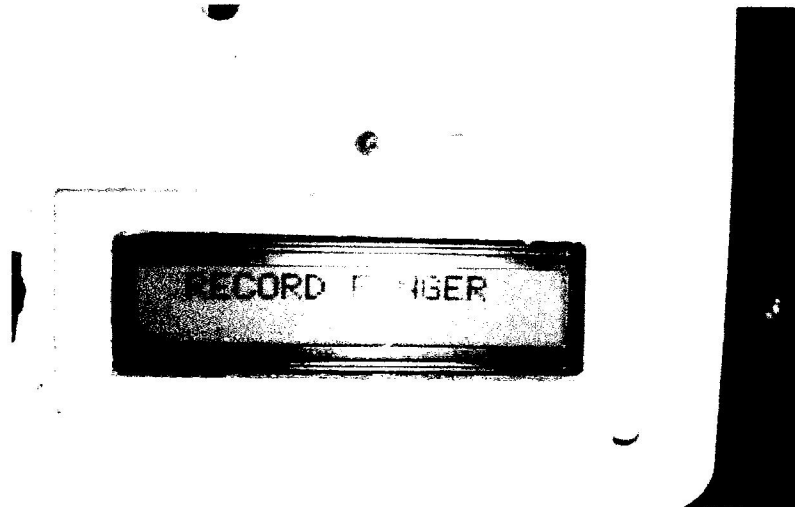




Fig 18: Fingerprint Enrollment

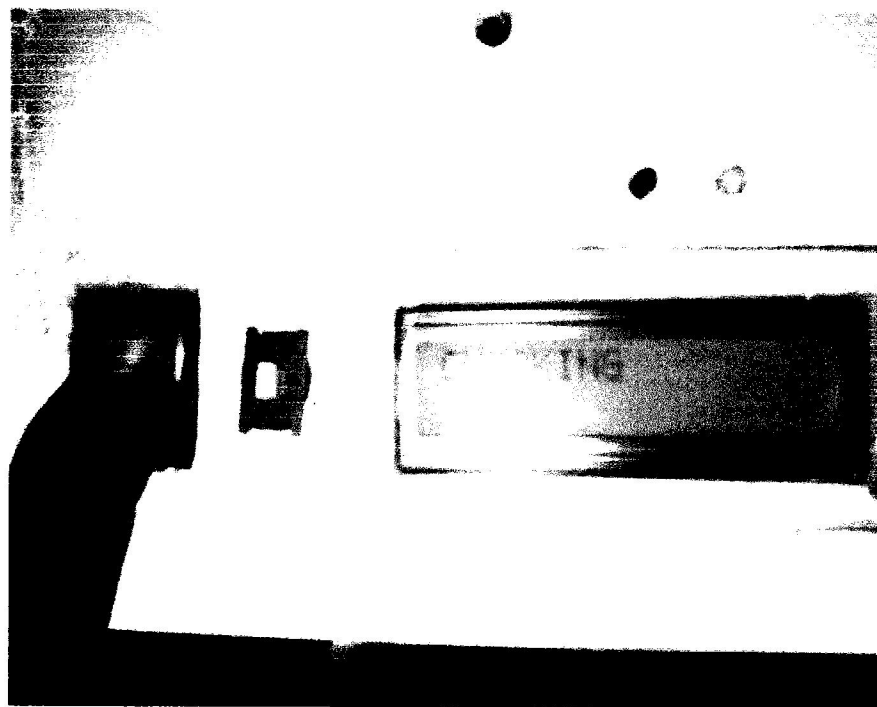


Fig 19: Fingerprint Matching

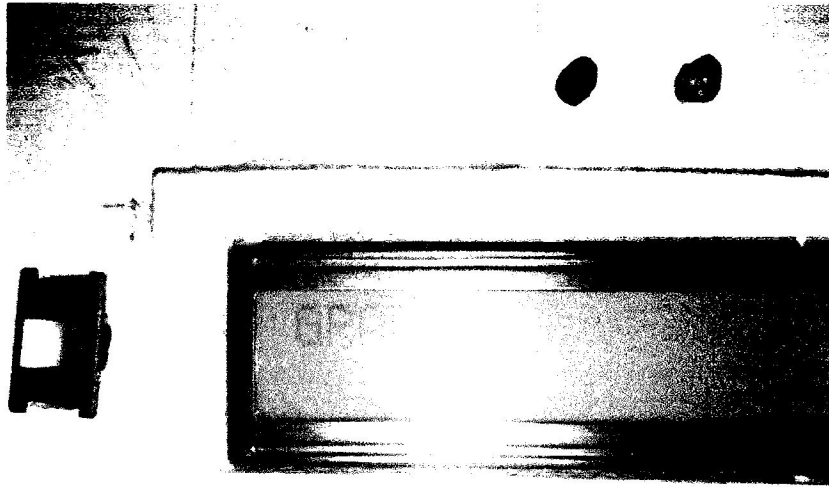


Fig 20: Fingerprint recognition (Authentication)

4.4 PROJECT MANAGEMENT

I chose the Gantt chart (as my project implementation analysis tool) to depict the schedule of implementation for the design work.

4.4.1. PROJECT SCHEDULE

Task	Start Date	End Date	Duration (Days)	Order
Project Simulation				
Simulation of the system	1/1/2020	1/15/2020	15	1
Simulation of the hardware	1/16/2020	1/30/2020	15	2
Simulation of the software	1/31/2020	2/14/2020	15	3
Simulation of the integration	2/15/2020	2/29/2020	15	4
Hardware Fabrication				
Hardware fabrication	3/1/2020	3/15/2020	15	5
Hardware testing	3/16/2020	3/30/2020	15	6
Soldering				
Soldering of components	4/1/2020	4/15/2020	15	7
Soldering of PCB	4/16/2020	4/30/2020	15	8
Soldering of connectors	5/1/2020	5/15/2020	15	9
Soldering of cables	5/16/2020	5/30/2020	15	10
Casing				
Casing of the device	6/1/2020	6/15/2020	15	11
Casing of the components	6/16/2020	6/30/2020	15	12
Casing of the cables	7/1/2020	7/15/2020	15	13

Fig 21: Table of Gantt chart

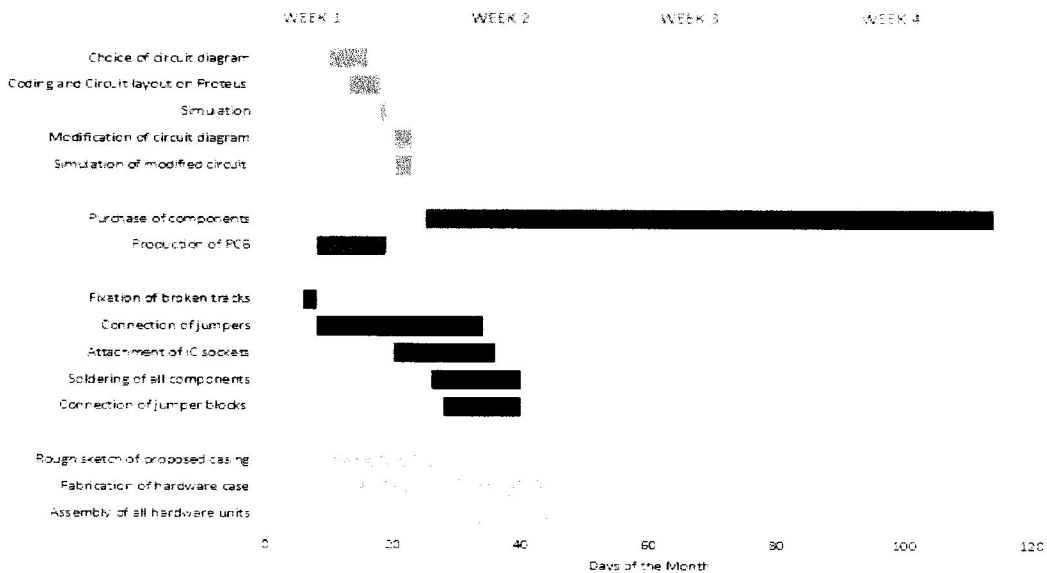


Fig 22: Project Implementation Schedule shown on a Gantt chart

4.5. RISK MANAGEMENT

One of the major risks taken in the course of this project was during the fabrication and production of PCB stage due to my inexperience in developing electronic circuit practically. This was then catered for by consulting various online platforms and engineers in the electronics field.

Another risk was for the fact that the serial communication ports were interchanged from the manufacturer which firstly led to misappropriations of the right wires to there various ports of the microcontroller.

Fearful risk attached to this project is that Attackers may chop off a victim's finger to get round a fingerprint recognition system used e.g. for the door. This attack is rather easy and highlights the need for effective liveliness detection and for other countermeasures such as multi-factor authentication. It also highlights the need for risk assessment: The deployment of a biometric system adds the users' limbs to the assets requiring protection and puts them at risk. Attackers may also try to activate a latent fingerprint on the sensor to work like a real finger by enforcing light, heat or moisture effects. The attack potentials for these attacks should be subject to further studies.

4.6. SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL CONSIDERATIONS

Biometric technology, like any other technology, suffers from unexpected and unforeseen consequences that many other technologies have experienced when implemented in society. Problems can arise when massive implementations are done. What happens when a biometric file

is stolen? A password or a credit card can be relatively easy replaced and the stolen information somehow invalidated. A biometric template is nothing more than another binary file in a database, therefore can be stolen by hackers as any other file. Losing our own biometrics may not be a matter of replacement; "with a biometric it is very difficult, if not impossible, for any individual to disassociate oneself from ones biometric. If biometric databases are not protected properly and information is stolen, the consequences can be permanently devastating. There is no easy way to program the biometric systems to not recognize a legit biometric of an authentic user. Democratic societies value individual privacy as well as government accountability. The rule of law shapes the manner by which the government may interact fairly with its citizenry¹⁰. The two fundamental legal principles related with Biometric technologies are the due process and the right of privacy:

"The concept of due process requires the government to acknowledge the possibility of errors, allowing means for their mitigation. The concept of privacy goes beyond simply acknowledging the possibility of errors to set limits on the power of government to meddle in the lives of individuals. These court protected guarantees require the government to respect the right of individuals by limiting intrusions to those which directly further recognizable and legitimate societal interest. This historical balance between individual rights and societal interest is at the heart of all democracies, and is placed under a new strain by the advent of biometric technologies"

ASEE has a strong recommendation about integrating ethics in engineering education¹⁵ either in each course or in a separate engineering course, but there is no specific number of hours recommended. The author recommends that engineering graduate and undergraduate courses in biometric security, automatic recognition and emerging technologies devote substantial time to discuss the social/legal consequences of biometric technologies. In my biometric courses, I invest about 5 hours (~10%) of the total course to discuss Ethics and legal consequences of biometric technologies. If the course is not directly related with biometric engineering and/or technology or additional knowledge needs to be induced in our students these can be done by means of homework and/or final papers. The author has implemented a five-hour ethics seminar in graduate courses like "Emerging Technologies" "Biometric security", "Global Technology" and an undergraduate course "Technology and Society". The seminar syllabus contains: A discussion of what biometric is and the core of the technology (i.e. recognition algorithms), practical implementation problems, problems with standards and enrollment, social and legal consequences of the technology with emphasis in the fourth, fifth and fourteen amendments of the US constitution.

CHAPTER FIVE

5.0. CONCLUSION AND RECOMMENDATION

Finally, the project fingerprint door based Authentication system has been successfully achieved. The need to have an automated door which gives entry via its fingerprint sensor helps to restrict unwanted personnel's into such rooms or buildings. This also adds security feature of a buzzer which notifies people around the building of an authorized persons trying to gain access after several times of trial.

This project works on the basis of registering the proposed fingerprints to be granted access using the fingerprint sensor, which is done by pressing the RECORD keypad until it displays RECORDED on the LCD. This fingerprints are stored into the sensors memory as templates. Any of the authorized user comes to access the door, places his registered finger on the sensor, the sensor then matches with the templates by displaying CHECKING, if found displays GRANTED and if not displays DENIED. If the unauthorized user tries access the door thrice, then BUZZER gives a loud sound creating some sort of awareness of an unauthorized user. Other modes of identification and communication which incorporate the disabled could also be implemented where modification on the mode of input and output can be implemented as an enhancement i.e. combining keypad with biometric identification or combining the LCD with an audio output.

In conclusion the use of the PIC16F778A microcontroller reduced the size and cost of implementing the circuit, making the digital lock system a low cost control system which could be used in homes, offices, schools and industrial premises.

5.1. CONTRIBUTION TO KNOWLEDGE

This work has majorly added two things which are:

- i. Security feature (buzzer): This feature which can also be referred to as an alarm feature gives chance for sensitivity or awareness of an unauthorized persons trying to gain access. This feature will help to reduce the distance, time administrative personnel should attend to the unauthorized persons prior to the use of gsm module which might have a delay in relating the message of an unauthorized persons to the administrative or security personnel's.
- ii. High sensitive biometric system (Fingerprint): This has always being one of the reliable means of biometrics as it gives room for accuracy, proper recognition and ease of authentication. This was further made possible with the finger print module which detects, verifies fingers with the already registered fingers in the module library. This module (fpm10a) is found to be less power consuming which makes it suitable to work at the least power supply.

5.2. LIMITATIONS

There are limitations to the project which are listed below:

- i. Firstly, the fingerprint module can't detect or verify a dirty finger which has been registered initially.
- ii. It has a capacity of 300 finger templates to be registered in its fingerprint library.
- iii. The sound made by the buzzer might not reach out to cover/ travel long distances to create some sorts of awareness or sensitization.
- iv. Fingerprints can be forged with the use of chemicals like cyanoacrylate, Ninhydrin etc. such chemicals help give chance to unauthorized persons. This is one of the limitations of this project but can be taken care of in further research by adding other biometric features like password, iris scanner etc.

5.3. FUTURE WORKS

Modifications can also be made to the whole system where information showing time, date and the user who opened the door could be sent and stored into a database. This would be used for monitoring personnel access to rooms within a building, where each personnel would have to input his/her user ID and a specific passcode for the ID. The system would then send the ID access information to a database system which would then store the information for analysis by management. Hence security and monitoring achieved. Further modifications that can be integrated to improve its features are:

- Iris scanner
- Password keypad
- GSM module (i.e. for faster security response)
- RFID reader

5.4. CRITICAL APPRAISAL

This design effectively created a lasting solution to the various issues encountered in the securing of important materials and it does encourage privacy. This design is found to be economical because of the low cost of components in achieving the required purpose. Due to the vast rise in technology, the fingerprints can be forged which therefore makes this project inefficient. This can be catered for by the integration of the iris technology which signifies and distinguishes various users from being authenticated.

BIBLIOGRAPHY AND REFERENCES

- [1]. Pavithra.B.C, "Fingerprint Based Bank Locker System Using Microcontroller", Proceedings of IRF International Conference, 05th April-2014, Pondicherry, India, ISBN: 978-93-82702-71-9
- [2]. Atar Nasrin, "Fingerprint Based Security System For Banks", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 –0056 Volume: 03 Issue: 04 | Apr-2016
- [3]. Omidiora E. O.(2011) "A Prototype of a Fingerprint Based Ignition Systems in Vehicles" Published in European Journal of Scientific Research ISSN 1450-216X Vol.62 No.2 (2011), pp. 164-171 © EuroJournals Publishing, Inc. 2011
- [4]. Sagar S. Palsodkar, "Bank Lockers Security System using Biometric and GSM Technology", SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) – Volume 2 Issue 4–April 2015
- [5]. The Biometric Consortium, "Introduction to Biometrics", (<http://www.biometrics.org>), 2006.
- [6]. "Fingerprint Matching" by Anil K. Jain, Jianjiang Feng and Karthik Nandakumar, Department of Computer Science and Engineering, Michign State University
- [7]. Mary Lourde R and Dushyant Khosla, "Fingerprint Identification in Biometric Security Systems", International Journal of Computer and Electrical Engineering, Vol. 2, No. 5, October, 2010.
- [8]. Hemant Kumar, "Fingerprint Based Locker with Image Capture", International Journal of Scientific Research and Management Studies (IJSRMS) ISSN: 2349-3771 Volume 3 Issue 5, pg: 198-202
- [9]. Raghu Ram Gangi, "Locker Opening And Closing System Using Rfid, Fingerprint, Password and GSM" International Journal of Emerging Trends & Technology in Computer Science (IJETTCS, Volume 2, Issue 2, March – April 2013 ISSN 2278-6856.
- [10]. Dr. Gerrit Hornung, LL.M., Monika Desoi, Matthias Pocs, LL.M. Biometric systems in future preventive Scenarios – Legal Issues and Challenges.
- [11]. Zhang Jinhai, Liu Xinjian, Chen Bo. The design and implementation of ID Authentication System Based on Fingerprint Identification.
- [12]. www.microchip.com

APPENDIX

APPENDIX I- GLOSSARY

RFID-radio frequency identification

GSM module-global system for mobile communication module.

PSU-power system unit.

APPENDIX II- TABLES

Table of fingerprint Module Pin Description

PIN NUMBER	Name	Type	Function Description
1	Vin	INPUT	Power Input.
2	GND	-	Signal ground connected to the power ground.
3	TD	INPUT	Data output. TTL logical level.
4	RD	OUTPUT	Out Data input. TTL logical level.

Electrical characteristics of LCD

Item	Symbol	Condition	Standard value			Unit	Applicable terminal
			Min.	Typ.	Max.		
Power voltage	V _{dd}		4.5	5.00	5.5	V	V _{dd}
Input High level voltage	V _{IH}		2.2		V _{dd}	V	RS, R/W, E, DB0, DB7
Input Low level voltage	V _{IL}		0.3	—	0.6	V	DB0, DB7
Output High level voltage	V _{OH}	I _{OH} = 0.205mA	2.4	—	—	V	DB0, DB7
Output Low level voltage	V _{OL}	I _{OL} = 1.8mA	—	—	0.4	V	DB0, DB7
I/O leakage current	I _{IL}	V _{in} = 0 ~ V _{dd}	-1	—	1.0	μA	RS, R/W, E, DB0, DB7
Supply current	I _{dd}	V _{dd} = 5V	8	—	—	mA	V _{dc}
LCD operating voltage	V _{LCD}	V _{dd} - V _O	3.0	—	1.0	V	V _O

APPENDIX II-PROGRAM CODE

INDF	EQU	0
TMR0	EQU	1
STATUS	EQU	3
FSR	EQU	4
TRISA	EQU	85H
TRISB	EQU	86H
TRISC	EQU	87H
TRISD	EQU	88H
TRISE	EQU	89H
PORTA	EQU	05H
PORTB	EQU	06H
PORTC	EQU	07H
PORTD	EQU	08H
PORTE	EQU	09H
ADCON1	EQU	9FH
ADCON0	EQU	1FH
ADRESL	EQU	9EH
ADRESH	EQU	1EH
INTCON	EQU	0BH
PIR1	EQU	0CH
PIE1	EQU	8CH
OPT	EQU	81H
TXREG	EQU	19H
RCREG	EQU	1AH
SPBRG	EQU	99H
TXSTA	EQU	98H
RCSTA	EQU	18H
TXIF	EQU	4
RCIF	EQU	5
EEDATA	EQU	10CH
EEADR	EQU	10DH
EEDATH	EQU	10EH
EEADRH	EQU	10FH
EECON1	EQU	18CH
EECON2	EQU	18DH
INTEMP	EQU	20H
INTEMPS	EQU	21H
DATA1	EQU	22H
DELAYV1	EQU	23H

INDF	EQU	0
TMR0	EQU	1
STATUS	EQU	3
FSR	EQU	4
TRISA	EQU	85H
TRISB	EQU	86H
TRISC	EQU	87H
TRISD	EQU	88H
TRISE	EQU	89H
PORTA	EQU	05H
PORTB	EQU	06H
PORTC	EQU	07H
PORTD	EQU	08H
PORTE	EQU	09H
ADCON1	EQU	9FH
ADCON0	EQU	1FH
ADRESL	EQU	9EH
ADRESH	EQU	1EH
INTCON	EQU	0BH
PIR1	EQU	0CH
PIE1	EQU	8CH
OPT	EQU	81H
TXREG	EQU	19H
RCREG	EQU	1AH
SPBRG	EQU	99H
TXSTA	EQU	98H
RCSTA	EQU	18H
TXIF	EQU	4
RCIF	EQU	5
EEDATA	EQU	10CH
EEADR	EQU	10DH
EEDATH	EQU	10EH
EEADRH	EQU	10FH
EECON1	EQU	18CH
EECON2	EQU	18DH
INTEMP	EQU	20H
INTEMPS	EQU	21H
DATA1	EQU	22H
DELAYV1	EQU	23H

```

DELAYV2 EQU    24H
DELAYV3 EQU    25H
COUNTER EQU    26H
RVALUE  EQU    27H
FOK     EQU    28H
DEGREE  EQU    29H
STATE   EQU    2AH
CSUM    EQU    2BH
STATE2  EQU    2CH
STOREADD EQU    EQU    70H
NEWT    EQU    71H
NEWT2   EQU    72H
ALARM   EQU    73H
        LIST   P=16F877
        ORG    0
        GOTO   START
        NOP
        NOP
        NOP
INT
        MOVWF  INTEMP
        SWAPF  STATUS,W
        MOVWF  INTEMPS

        SWAPF  INTEMPS,W
        MOVWF  STATUS
        SWAPF  INTEMP,F
        SWAPF  INTEMP,W
        RETFIE
SERIAL1
        CLRF   RCSTA
        BANKSEL SPBRG
        MOVLW 0X15 ; baudrate fosc/16/(0x15+1)
        MOVWF  SPBRG ;
        MOVLW 0X24
        MOVWF  TXSTA
        BANKSEL RCSTA
        MOVLW 0X90
        MOVWF  RCSTA

```



```

                RETURN
SERIAL2
    BTFSS    PIR1, TXIF
    GOTO    SERIAL2
    MOVWF   TXREG
    RETURN

SERIAL3
    BTFSS    PIR1, RCIF
    GOTO    SERIAL3
    MOVF    RCREG, W
    MOVWF   RVALUE
    RETURN

GENFINGER
    MOVLW   0XEF
    CALL    SERIAL2
    MOVLW   0X01
    CALL    SERIAL2
    MOVLW   0XFF
    CALL    SERIAL2
    MOVLW   0XFF
    CALL    SERIAL2
    MOVLW   0XFF
    CALL    SERIAL2
    MOVLW   0XFF
    CALL    SERIAL2
    MOVLW   0X01
    CALL    SERIAL2
    RETURN

RECORDF
    CALL    GENFINGER
    MOVLW   0X00
    CALL    SERIAL2
    MOVLW   0X03
    CALL    SERIAL2
    MOVLW   0X01
    CALL    SERIAL2
    MOVLW   0X00
    CALL    SERIAL2
    MOVLW   0X05

```

```

        CALL    SERIAL2
        RETURN
FEATUREF
        CALL    GENFINGER
        MOVLW  0X00
        CALL    SERIAL2
        MOVLW  0X04
        CALL    SERIAL2
        MOVLW  0X02
        CALL    SERIAL2
        MOVLW  0X01
        CALL    SERIAL2
        MOVLW  0X00
        CALL    SERIAL2
        MOVLW  0X08
        CALL    SERIAL2
        RETURN
STOREF
        MOVLW  0X0E
        ADDWF  STOREADD,W
        MOVWF  CSUM
        CALL    GENFINGER
        MOVLW  0X00
        CALL    SERIAL2
        MOVLW  0X06
        CALL    SERIAL2
        MOVLW  0X06
        CALL    SERIAL2
        MOVLW  0X01
        CALL    SERIAL2
        MOVLW  0X00
        CALL    SERIAL2
        MOVF   STOREADD,W
        CALL    SERIAL2
        MOVLW  0X00
        CALL    SERIAL2
        MOVF   CSUM,W
        CALL    SERIAL2
        RETURN

```


SEARCHF

```
CALL    GENFINGER
MOVLW  0X00
CALL    SERIAL2
MOVLW  0X08
CALL    SERIAL2
MOVLW  0X04
CALL    SERIAL2
MOVLW  0X01
CALL    SERIAL2
MOVLW  0X00
CALL    SERIAL2
MOVLW  0X00
CALL    SERIAL2
MOVLW  0X00
CALL    SERIAL2
MOVLW  0X80
CALL    SERIAL2
MOVLW  0X00
CALL    SERIAL2
MOVLW  0X8E
CALL    SERIAL2
RETURN
```

EMPTYF

```
CALL    GENFINGER
MOVLW  0X00
CALL    SERIAL2
MOVLW  0X03
CALL    SERIAL2
MOVLW  0X0D
CALL    SERIAL2
MOVLW  0X00
CALL    SERIAL2
MOVLW  0X11
CALL    SERIAL2
RETURN
```

PROCESSF1

```
CLRF   FOK
CALL   SERIAL3
```

```

MOVF    RVALUE,W
SUBLW   0XEF
BTSS    STATUS,2
GOTO    ERROR1
CALL    SERIAL3
MOVF    RVALUE,W
SUBLW   0X01
BTSS    STATUS,2
GOTO    ERROR1
CALL    SERIAL3
MOVF    RVALUE,W
SUBLW   0XFF
BTSS    STATUS,2
GOTO    ERROR1
CALL    SERIAL3
MOVF    RVALUE,W
SUBLW   0XFF
BTSS    STATUS,2
GOTO    ERROR1
CALL    SERIAL3
MOVF    RVALUE,W
SUBLW   0XFF
BTSS    STATUS,2
GOTO    ERROR1
CALL    SERIAL3
MOVF    RVALUE,W
SUBLW   0X07
BTSS    STATUS,2
GOTO    ERROR1
RETURN
PROCESSF2
CALL    SERIAL3
MOVF    RVALUE,W
SUBLW   0X00

```

```

        BTFS    STATUS,2
        GOTO   ERROR1
        CALL   SERIAL3
        MOVF   RVALUE,W
        SUBLW  0X07
        BTFS   STATUS,2
        GOTO   ERROR1
        RETURN

ERROR1
        MOVLW  0X02
        MOVWF  FOK
        RETURN

MODE1ER
        BTFS   NEWT2,7
        RETURN
        CALL   LCDCLR
        CALL   LCDU
        CALL   SPP
        CALL   SPP
        CALL   SPP
        CALL   SPP
        CALL   RET
        RETURN

DSPEXIST
        CALL   LCDCLR
        CALL   LCDU
        CALL   SPP
        CALL   SPP
        CALL   SPP
        CALL   SPP
        CALL   SPP
        CALL   EXI
        MOVLW  0XFF
        MOVWF  NEWT
        MOVWF  NEWT2
        RETURN

FINGEROK
FEND
        CALL   SERIAL3
        CALL   SERIAL3

```

```

RETURN
PROCESSF3
CALL    SERIAL3
MOVWF  RVALUE,W
SUBLW  0X00
BTFSC  STATUS,2
GOTO   FINGEROK
MOVLW  0X01
MOVWF  FOK
GOTO   FEND

PROCESSF3B
MOVWF  RVALUE,W
MOVWF  DEGREE
CALL   SERIAL3
CALL   SERIAL3
GOTO   FEND

```

```

MODEGEN
CALL   RECORDF
CALL   PROCESSF1
BTFSC  FOK,1
GOTO   MODE1ER
CALL   PROCESSF2
BTFSC  FOK,1
GOTO   MODE1ER
CALL   PROCESSF3
BTFSC  FOK,0
GOTO   MODE1ER
CALL   FEATUREF
CALL   PROCESSF1
BTFSC  FOK,1
GOTO   MODE1ER
CALL   PROCESSF2
BTFSC  FOK,1
GOTO   MODE1ER
CALL   PROCESSF3

```

```

    BTFSC    FOK,0
    GOTO     MODE1ER
    CALL     SEARCHF
    CALL     PROCESSF1
    BTFSC    FOK,1
    GOTO     MODE1ER
    CALL     PROCESSF2B
    BTFSC    FOK,1
    GOTO     MODE1ER
    CALL     PROCESSF3
    CALL     PROCESSF3B
    BTFSS    FOK,0
    GOTO     DSPEXIST
    BTFSC    STATE2,0
    GOTO     MGEN2B

MGEN2
    CALL     STOREF
    CALL     PROCESSF1
    BTFSC    FOK,1
    GOTO     MODE1ER
    CALL     PROCESSF2
    BTFSC    FOK,1
    GOTO     MODE1ER
    CALL     PROCESSF3
    BTFSC    FOK,0
    GOTO     MODE1ER

MGEN2B
    MOVLW   0X01
    MOVWF   STATE
    RETURN

|
EREAD
    BANKSEL EEADR
    CLRF   EEADR
    BANKSEL EECON1
    BCF    EECON1,7
    BSF    EECON1,0
    BANKSEL EEADR
    MOVF   EEDATA,W

```

```

MOVWF STOREADD
BANKSEL PORTA
RETURN
EWRITE
BANKSEL EEADR
CLRF EEADR
MOVF STOREADD,w
MOVWF EEDATA
BANKSEL EECON1
BCF EECON1,7
BSF EECON1,2
BCF INTCON,7
MOVLW 0x55
MOVWF EECON2
MOVLW 0xAA
MOVWF EECON2
BSF EECON1,1
BSF INTCON,7
BCF EECON1,2
WP1
BTFSC EECON1,1
GOTO WP1
BANKSEL PORTA
RETURN
ADC1
BANKSEL EECON1
BCF EECON1,7
BANKSEL PORTA
CLRF PORTA
MOVLW 0xFF
MOVWF PORTB
CLRF PORTC
CLRF PORTD
CLRF PORTE
BANKSEL TRISA
CLRF TRISA
MOVLW 0xC0
MOVWF TRISC
CLRF TRISD

```

```

        CLRF    TRISE
        MOVLW  0X06
        MOVWF  ADCON1
        CLRF   OPT
        BANKSEL PORTA
        MOVLW  0X20
        MOVWF  FSR

FNEXT
        CLRF   INDF
        INCF   FSR, F
        MOVF   FSR, W
        SUBLW  0X7F
        BTFSS  STATUS, 2
        GOTO   FNEXT
        CALL   INTLCD
        MOVLW  0X02
        MOVWF  PORTA
        CALL   SERIAL1
        RETURN

DELAY1SEC
        CLRF   DELAYV1
        CLRF   DELAYV2
        CLRF   DELAYV3
        BSF    DELAYV3, 0
        BSF    DELAYV3, 2
        BSF    DELAYV3, 4

DELAYB
        DECFSZ DELAYV1, F
        GOTO   DELAYB
        DECFSZ DELAYV2, F
        GOTO   DELAYB
        DECFSZ DELAYV3, F
        GOTO   DELAYB
        RETURN

RSLOW
        CLRF   PORTE
        MOVLW  0X02
        MOVWF  PORTE
        MOVF   DATA1, W

```

```

MOVWF PORTD
NOP
CLRF PORTE
MOVLW 0X32
CALL DELAY
RETURN
RSHIGH
MOVLW 0X01
MOVWF PORTE
MOVLW 0X03
MOVWF PORTE
MOVF DATA1,W
MOVWF PORTD
MOVLW 0X01
MOVWF PORTE
MOVLW 0X32
CALL DELAY
RETURN
DELAY
MOVWF COUNTER
DELAY2
NOP
NOP
DECFSZ COUNTER,F
GOTO DELAY2
RETURN
INTLCD
CALL DELAY1SEC
MOVLW 0X3C
MOVWF DATA1
CALL RSLW
MOVLW 0X06
MOVWF DATA1
CALL RSLW
MOVLW 0X0C
MOVWF DATA1
CALL RSLW
MOVLW 0X80
MOVWF DATA1

```



```
CALL    RSLW
LCDCLR  MOVW  0X01
        MOVW  DATA1
        CALL  RSLW
        CALL  DELAY1SEC
        RETURN
```

```
DEL     MOVW  0X44
        MOVW  DATA1
        CALL  RSHIGH
        MOVW  0X45
        MOVW  DATA1
        CALL  RSHIGH
        MOVW  0X4C
        MOVW  DATA1
        CALL  RSHIGH
        MOVW  0X45
        MOVW  DATA1
        CALL  RSHIGH
        MOVW  0X54
        MOVW  DATA1
        CALL  RSHIGH
        RETURN
```

```
EEE     MOVW  0X45
        MOVW  DATA1
        CALL  RSHIGH
        RETURN
```

```
PLAY   MOVW  0X50
        MOVW  DATA1
        CALL  RSHIGH
        MOVW  0X4C
        MOVW  DATA1
        CALL  RSHIGH
        MOVW  0X41
        MOVW  DATA1
        CALL  RSHIGH
```

	MOVLW	0X59
	MOVWF	DATA1
	CALL	RSHIGH
	RETURN	
PLA		
	MOVLW	0X50
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X4C
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X41
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X43
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X45
	MOVWF	DATA1
	CALL	RSHIGH
	RETURN	
AAA		
	MOVLW	0X41
	MOVWF	DATA1
	CALL	RSHIGH
	RETURN	
BBB		
	MOVLW	0X42
	MOVWF	DATA1
	CALL	RSHIGH
	RETURN	
SPP		
	MOVLW	0X20
	MOVWF	DATA1
	CALL	RSHIGH
	RETURN	
FIN		
	MOVLW	0X46
	MOVWF	DATA1

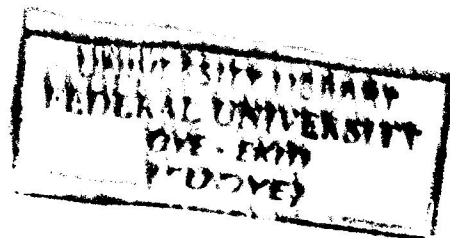
```
CALL    RSHIGH
MOVLW  0X49
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X4E
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X47
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X45
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X52
MOVWF  DATA1
CALL    RSHIGH
RETURN
```

RECC

```
MOVLW  0X52
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X45
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X43
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X20
MOVWF  DATA1
CALL    RSHIGH
RETURN
```

REC

```
MOVLW  0X52
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X45
MOVWF  DATA1
CALL    RSHIGH
MOVLW  0X43
```



```

MOVWF DATA1
CALL RSHIGH
MOVLW 0X4F
MOVWF DATA1
CALL RSHIGH
MOVLW 0X52
MOVWF DATA1
CALL RSHIGH
MOVLW 0X44
MOVWF DATA1
CALL RSHIGH
RETURN

AND
MOVLW 0X41
MOVWF DATA1
CALL RSHIGH
MOVLW 0X4E
MOVWF DATA1
CALL RSHIGH
MOVLW 0X44
MOVWF DATA1
CALL RSHIGH
RETURN

PRE
MOVLW 0X50
MOVWF DATA1
CALL RSHIGH
MOVLW 0X52
MOVWF DATA1
CALL RSHIGH
MOVLW 0X45
MOVWF DATA1
CALL RSHIGH
MOVLW 0X53
MOVWF DATA1
CALL RSHIGH
MOVLW 0X53
MOVWF DATA1
CALL RSHIGH

```

```

RETURN
OKK
    MOVLW    0X4F
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X4B
    MOVWF    DATA1
    CALL     RSHIGH
    RETURN

EXI
    MOVLW    0X45
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X58
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X49
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X53
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X54
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X53
    MOVWF    DATA1
    CALL     RSHIGH
    RETURN

VOI
    MOVLW    0X56
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X4F
    MOVWF    DATA1
    CALL     RSHIGH
    MOVLW    0X49
    MOVWF    DATA1
    CALL     RSHIGH

```

```

    MOVLW 0X43
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X45
    MOVWF DATA1
    CALL RSHIGH
    RETURN
RET
    MOVLW 0X52
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X45
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X54
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X52
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X59
    MOVWF DATA1
    CALL RSHIGH
    RETURN
GRA
    MOVLW 0X47
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X52
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X41
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X4E
    MOVWF DATA1
    CALL RSHIGH
    MOVLW 0X54
    MOVWF DATA1

```

	CALL	RSHIGH
	RETURN	
DEN	MOVLW	0X44
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X45
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X4E
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X49
	MOVWF	DATA1
	CALL	RSHIGH
	RETURN	
ING	MOVLW	0X49
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X4E
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X47
	MOVWF	DATA1
	CALL	RSHIGH
	RETURN	
EDD	MOVLW	0X45
	MOVWF	DATA1
	CALL	RSHIGH
	MOVLW	0X44
	MOVWF	DATA1
	CALL	RSHIGH
	CALL	SPP
	RETURN	
LCDU	MOVLW	0X80
	MOVWF	DATA1

```

MOVWF DATA1
CALL RSLOW
RETURN
LCDL
MOVLW 0XC0
MOVWF DATA1
CALL RSLOW
RETURN
DSPCHECKING
CALL LCDCLR
CALL LCDU
MOVLW 0X43
MOVWF DATA1
CALL RSHIGH
MOVLW 0X48
MOVWF DATA1
CALL RSHIGH
MOVLW 0X45
MOVWF DATA1
CALL RSHIGH
MOVLW 0X43
MOVWF DATA1
CALL RSHIGH
MOVLW 0X4B
MOVWF DATA1
CALL RSHIGH
MOVLW 0X49
MOVWF DATA1
CALL RSHIGH
MOVLW 0X4E
MOVWF DATA1
CALL RSHIGH
MOVLW 0X47
MOVWF DATA1
CALL RSHIGH
RETURN
RECORDFINGERA
CALL LCDCLR

```



```

CALL    SPP
CALL    REC
CALL    SPP
CALL    FIN
CALL    DELAY1SEC
RETURN
RECORDFINGER
CLRF    NEWT
CLRF    NEWT2
CALL    SERIAL1
CALL    LCDCLR
CALL    LCDU
CALL    PLA
CALL    SPP
CALL    AAA
CALL    SPP
CALL    FIN
CALL    LCDL
CALL    SPP
CALL    AND
CALL    SPP
CALL    PRE
CALL    SPP
CALL    OKK
RFLOOP
BTFS    PORTB,2
GOTO    RFLOOP
CALL    LCDCLR
CALL    LCDU
CALL    SPP
CALL    SPP
CALL    SPP
CALL    SPP
CALL    REC
CALL    ING
SUBNEWT
CLRF    STATE
CALL    MODEGEN
BTFS    STATE,0

```

```

        GOTO    SUBNEW2
        CALL    LCDCLR
        CALL    LCDU
        CALL    SPP
        CALL    SPP
        CALL    SPP
        CALL    SPP
        CALL    REC
        CALL    EDD
        INCF    STOREADD, F
        CALL    EWRITE
        CALL    DELAY1SEC
        CALL    DELAY1SEC
        CALL    RECORDFINGERA
        MOVLW   0xFF
        MOVWF   NEWT
        MOVWF   NEWT2
        RETURN

SUBNEW2
        BTFS   NEWT2, 7
        RETURN
        INCF   NEWT, F
        MOVF   NEWT, W
        BTFS   STATUS, 2
        GOTO   SUBNEW2B
        INCF   NEWT2, F

SUBNEW2B
        BTFS   NEWT2, 2
        GOTO   SUBNEW
        CALL   LCDCLR
        CALL   LCDU
        CALL   SPP
        CALL   SPP
        CALL   SPP
        CALL   SPP
        CALL   RET
        CALL   DELAY1SEC
        RETURN

DELETE

```

```

CALL SERIAL1
CALL LCDCLR
CALL LCDU
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL DEL
CALL EEE

DLLOOP
BTFSK PORTB,2
GOTO DLLOOP
CALL LCDU
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL DEL
CALL ING
CALL EMPTYF
CALL PROCESSF1
BTFSK FOK,1
GOTO MODE1ER
CALL PROCESSF2
BTFSK FOK,1
GOTO MODE1ER
CALL PROCESSF3
BTFSK FOK,0
GOTO MODE1ER
CLRF STOREADD
CALL EWRITE
CALL LCDU
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP

```

```

CALL DEL
CALL EDD
CALL DELAY1SEC
CALL DELAY1SEC
CALL RECORDFINGERA
RETURN
START
CALL ADC1
BEGIN
BTFS PORTB, 0
GOTO BEGIN2
CALL EREAD
CALL RECORDFINGERA
BEGINB
BTFS PORTB, 2
CALL RECORDFINGER
BTFS PORTB, 7
CALL DELETE
GOTO BEGINB
BEGIN2
BTFS PORTB, 1
GOTO BEGIN3
CALL LCDCLR
CALL LCDU
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL VOI
BEGIN2B
BTFS PORTB, 3
GOTO PLAYA
BTFS PORTB, 4
GOTO PLAYB
BTFS PORTB, 5
GOTO RECA
BTFS PORTB, 6

```

```

        GOTO    RECB
        GOTO    BEGIN2B
PLAYA
        CALL    LCDL
        CALL    PLAY
        CALL    SPP
        CALL    AAA
        MOVLW  0X02
        MOVWF  PORTC
        GOTO    BEGIN2A
PLAYB
        CALL    LCDL
        CALL    PLAY
        CALL    SPP
        CALL    BBB
        MOVLW  0X04
        MOVWF  PORTC
        GOTO    BEGIN2A
RECA
        CALL    LCDL
        CALL    RECC
        CALL    SPP
        CALL    AAA
        MOVLW  0X03
        MOVWF  PORTC
        GOTO    BEGIN2A
RECB
        CALL    LCDL
        CALL    RECC
        CALL    SPP
        CALL    BBB
        MOVLW  0X05
        MOVWF  PORTC
        GOTO    BEGIN2A
BEGIN2A
        CALL    DELAY1SEC
        CLRF   PORTC
        CALL    DELAY1SEC
        CALL    LCDL

```

```

CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
CALL SPP
GOTO BEGIN2B

BEGIN3
MOVLW 0X01
MOVWF STATE2
CLRF PORTC
CALL LCDCLR
CALL LCDU
CALL PLA
CALL SPP
CALL AAA
CALL SPP
CALL FIN
CALL LCDL
CALL SPP
CALL AND
CALL SPP
CALL PRE
CALL SPP
CALL OKK

PFLOOP
BTFSK PORTB,2 ;ok
GOTO PFLOOP
CLRF NEWT
CLRF NEWT2
CALL DSPCHECKING

BEGIN444
CLRF STATE
CALL MODEGEN ;; v.i
BTFSK STATE,0
GOTO BEGIN4

```

```

BTFSK    FOK,0
GOTO     BEGIN4
MOVLW    0X02
MOVWF    PORTC
CALL     LCDCLR
CALL     LCDU
CALL     GRA
CALL     EDD
MOVLW    0X01
MOVWF    PORTA
CALL     DELAY1SEC
CALL     DELAY1SEC
CLRF     PORTA
CLRF     ALARM
GOTO     BEGIN3

BEGIN4
INCF     NEWT, F
MOVF     NEWT, W
BTFSK    STATUS, 2
GOTO     BEGIN44
INCF     NEWT2, F

BEGIN44
BTFSK    NEWT2, 1
GOTO     BEGIN444

MOVLW    0X04
MOVWF    PORTC
CALL     LCDCLR
CALL     LCDU
CALL     DEN
CALL     EDD

MOVF     ALARM, W
SUBLW    0X03
BTFSK    STATUS, 2
INCF     ALARM, F

MOVF     ALARM, W
SUBLW    0X03

```

CALL SUBALARM

CALL DELAY1SEC

CALL DELAY1SEC

GOTO BEGIN3

SUBALARM

MOVLW 0X05

MOVWF PORTC

CALL DELAY1SEC

CALL DELAY1SEC

CALL DELAY1SEC

RETURN

END



**DESIGN AND IMPLEMENTATION OF A FINGER-
PRINT DOOR BASED AUTHENTICATION SYSTEM**

BY

OJO OLUWATOBILOBA JOHNSON

(MATRIC NO: EEE/13/1113)

**A PROJECT REPORT SUBMITTED TO THE DEPARTMENT
OF ELECTRICAL AND ELECTRONICS ENGINEERING,
FEDERAL UNIVERSITY OYE-EKITI, IKOLE EKITI
CAMPUS.**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
AWARD OF BACHELOR OF ENGINEERING (B.ENG) IN
ELECTRICAL AND ELECTRONICS ENGINEERING.**

FEBRUARY, 2019.

