UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

FINGERPRINT BASED DOOR LOCK SYSTEM

**M. Sc. THESIS
IN
ELECTRICAL AND ELECTRONICS ENGINEERING**

BY

MEHMET MERKEPÇİ

SEPTEMBER 2009

Fingerprint Based Door Lock System

M.Sc. Thesis
in
Electrical and Electronics Engineering
University of Gaziantep

Supervisor:
Prof. Dr. M.Sadettin ÖZYAZICI

by

Mehmet MERKEPÇİ
September 2009

T.C.
UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL&APPLIED SCIENCES
ELECTRICAL&ELECTRONICS ENGINEERING

Name of the thesis : Fingerprint based door lock system.

Name of the student: Mehmet MERKEPÇİ

Exam date : 01.09.2009

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Ramazan KOÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Savaş UÇKUN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. M.Sadettin ÖZYAZICI
Supervisor

Examining Committee Members Signature

1. Prof. Dr. Ramazan KOÇ..__________
2. Prof. Dr. Arif NACAROĞLU..__________
3. Prof. Dr. M.Sadettin ÖZYAZICI..__________

ABSTRACT

FINGERPRINT BASED DOOR LOCK SYTEM

Mehmet MERKEPÇİ
M.Sc. in Electrical & Electronics Eng.
Supervisor: Prof. Dr. M.Sadettin ÖZYAZICI
September 2009, (pages 79)

Popular biometric systems in use today include iris recognition, voice recognition, and fingerprint recognition systems. Iris recognition is extremely accurate but expensive to implement, and scanning the human eye is a sensitive issue. A typical voice recognition system is much less expensive but often exhibits unacceptably hoarseness or other throat problems. Fingerprint recognition is generally considered the most practical choice for its reliability, non-intrusive interfaces, and cost-effectiveness. Biometric fingerprint door locks have lots of advantages over conventional key door locks, keyless keypad lock or card reader door locks.

Thus, biometric fingerprint door locks predominate to security protection, convenience, and speed. So, fingerprint based systems are the most convenience and practical of access control systems used in the door lock system.

The companies are using such a system for controlling the employers. Their staff must have the conventional card and ill intentioned people can read the cards instead of each other. Each person's fingerprint is unique and the staff do not have a chance to register the fingerprint instead of each other, the system operates smoothly and prevents fraud.

Therefore, in this study, personnel access control and time attendance system which is based on a fingerprint has been investigated.

Key words: fingerprint, access control, door lock, time attendance

ÖZET

PARMAK İZİNE DAYALI KAPI KİLİT SİSTEMİ

Mehmet MERKEPÇİ
Yüksek Lisans Tezi, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. M. Sadettin ÖZYAZICI

Eylül 2009 (79 sayfa)

Günümüzde kullanılan popüler biyometrik sistemleri genelde iris tanıma, ses tanıma ve parmakizi tanıma sistemlerinde oluşmaktadır. İris tanıma son derece doğru fakat pahalı ve aynı zamanda hassas bir sistemdir. Tipik bir ses tanıma sistemi daha ucuz bir sistem olmasına rağmen ses kısıklığı ve diğer boğaz sorunları gibi problemler yüzünden sistem güvenilirliğini yitirir. Parmakizi tanıma is bu sistemler arasında kullanışlı arayüzü, etkin maliyeti, pratikliği ve güvenilirliği açısından en uygun ve en fazla kullanılan sistemdir. Biyometrik parmak izi kapı kilitleri geleneksel anahtar kapı kilitleri, anahtarsız tuş kilidi veya kart okuyuculu kapı kilitleri gibi sistemlere göre daha fazla avantajlı olduğu oldukça açıktır. Bunun için biyometrik kapı kilitleri, güvenlik koruması, kolaylığı ve hızı bakımından diğer sistemlere nazaran daha üstündür. Bu özellikleri bakımından parmak izi geçiş kontrol sistemleri günümüzde yaygın olarak kullanılmaktadır.

Şirketlerde böyle bir sistemin kullanılması ise işveren açısından çok önemli avantajlara sahiptir. Klasik kartlı geçiş sistemlerinde her personele ait bir kart bulunması gerekir ve kötü niyetli kişiler kartlarını birbirinin yerine okutabilir. Her kişinin parmak izi tek olduğu ve personellerin birbiri yerine parmaklarını okutma şansı olmadığı için sistem sorunsuz çalışır ve sahteciliği önler. Dolayısıyla bu çalışmada parmak izine dayalı personel geçiş ve devam kontrol sistemi üzerine bir çalışma yapılmıştır.

Anahtar Kelimeler: Parmakizi, geçiş kontrol, kapı kilit, devam kontrol

ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude thanks to my supervisor, Prof. Dr. M.Sadettin ÖZYAZICI for his advice and guidance in the preparation of this thesis.

Also, I would like to express my sincere gratitude to my wife Hamiyet MERKEPÇİ and my sister Seval UYANIK for their grand support and patience, confidence for my higher education.

Lastly, I wish to special thank my parents.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

TIME ATTENDANCE SYSTEMS

CHAPTER 3

FINGERPRINT CLASSIFICATION AND MATCHING

CHAPTER 4

DESIGN AND CONSTURUCTION OF FINGERPRINT SYTEM

CHAPTER 5

PROGRAMMING OF ACCESS CONTROL SYSTEM

CHAPTER 6

APPENDICES

APPENDIX A

APPENDIX B

APPENDIX C

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

The following nomenclature defines the principal symbol used in the thesis.

Symbols	**Description**
NIST	National Institute of Standards and Technology
FBI	Federal Bureau of Investigation
NBS	National Bureau of Standards
AFIS	Automated Fingerprint Identification Systems
IAFIS	Integrated Automated Fingerprint Identification System
IBM	International Business Machines
IVR	Interactive Voice Response
WIFI	Wireless Fidelity
IT	Information Technology
AC&TA	Access Control and Time Attendance
C#	Visual C Sharp
DBMS	Database Management Systems
FTIR	Frustrated Total Internal Reflection

CHAPTER 1

INTRODUCTION

Fingerprint based systems are widely used everywhere.

1.1 Introduction

Fingerprint recognition is the most important part of biometric. Biometric is the most popular method of physical details belonging to a person with the aim of identification in a reliable way [1]. A biometric system includes many personal biological characteristics that are now used for personal recognition. For example, iris recognition, fingerprint recognition, voice recognition, signature recognition or face recognition. Since biological features are unique, and thus more reliable to identify people than traditional methods, such systems are more useful and suitable than traditional procedures. The traditional methods are commonly based on features that we have (key, card) or we know (password).

Biometric systems are now commonly used in different parts of everyday life such as building access and computer login. Fingerprint recognition is the most widely used for personal identification within all biometric systems [2]. Throughout the world, fingerprint recognition is accepted by a large part of the population because of its fast, secure, and easy way of personal identification.

Typically, fingerprints consist of ridge and valley patterns on the tips of human fingers. Thanks to their uniqueness and continuity, the use of fingerprints is considered to be one of the most reliable methods of personal verification. “Due to the continuing needs of law enforcement and interest from the developers of civilian applications, automated fingerprint verification system are becoming increasingly widespread and are being extensively researched by the pattern recognition and image processing communities” [3]. And also fingerprint recognition is today being increasingly used in a large number of various applications such as access control or online identification [4].

1.2 History of Fingerprint

Fingerprints have been used for a very long time for personal identification. Fingerprint matching algorithm techniques were started in the 16th century. Henry Fauld, in 1880, suggested individuality and uniqueness of fingerprint. Also, Herschel claimed that he practiced fingerprint identification for many years. This invention established the foundation of modern fingerprint identification. In the 19th century, Francis Galton conducted a common study of fingerprints [5]. Francis Galton defined some of the points or characteristics from which fingerprints can be identified. These "Galton Points" are the foundation for the science of fingerprint identification, which has expanded and transitioned over the past century. Fingerprint identification began its transition to automation in the late 1960s along with the emergence of computing technologies. With the advent of computers, a subset of the Galton Points, referred to as minutiae, has been utilized to develop automated fingerprint technology.

He presented the minutiae features for fingerprint classification in 1888. The discovery of uniqueness of fingerprints caused an immediate decline in the prevalent use of anthropometric methods of identification and led to the appropriation of fingerprints as a more efficient method of identification [6].

In the late of 1960, Federal Bureau of Investigation (FBI) develops a system**.** Fingerprint identification process is automatically done in this system. The FBI accepted the National Bureau of Standards (NBS), now the National Institute of Standards and Technology (NIST), to study the process of automating fingerprint classification, searching, and matching**.** System based on two methods**.**

- Fingerprint scanning and extracting minutiae from each fingerprint.
- Searching, comparing, and matching lists of minutiae in large stores of fingerprints.

In 1975, the FBI decided to support the development of fingerprint scanners for automated classifiers and minutiae extraction technology. Owing to this support a prototype reader was developed. To collect the fingerprint minutia, this early reader used capacitive techniques. At the same time, singular biographical data, fingerprint classification data, and minutiae were stored because the price of storage for the digital images of the fingerprints was expensive.

NIST, in order to develop automatic methods of digitizing inked fingerprint, put a great deal of effort. Over the next few decades, NIST pioneered the efforts which developed methods of image compression on image quality, classification, extraction of minutiae, and matching. At the end of the works, eventually, NIST developed the M40 algorithm. It obtained the first operational matching algorithm and used it at the FBI for the human search. The results produced by the M40 algorithm were provided to trained and specialized human technicians who evaluated the significantly smaller set of candidate images. The available fingerprint technology continued to improve and by 1981, five Automated Fingerprint Identification Systems (AFIS) had been deployed. Various state systems within the US and other countries had implemented their own standalone systems, developed by a number of different vendors. During this evolution, communication and information exchange between the systems were overlooked, meaning that a fingerprint collected on one system could not be searched against another system. These oversights led to the need for and development of fingerprint standards.

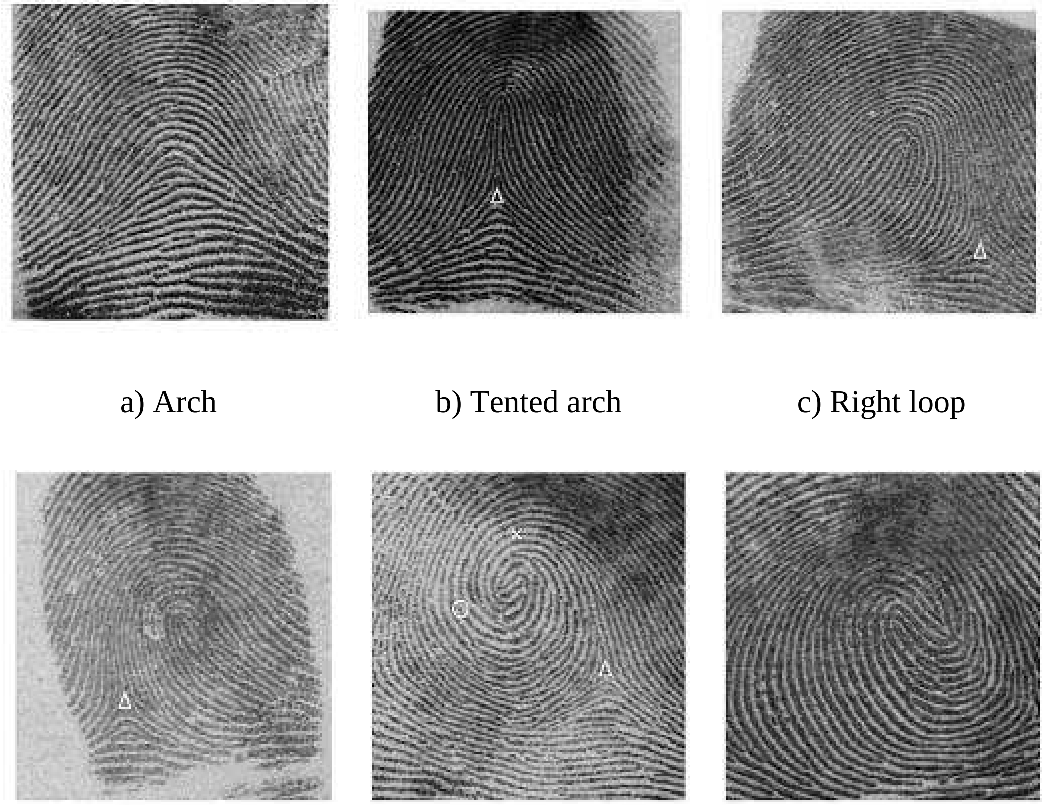
As the need for an integrated identification system within the US criminal justice community quickly became apparent, the next stage in fingerprint automation occurred at the end of the Integrated Automated Fingerprint Identification System (IAFIS) competition in 1994. The competition identified and investigated three major challenges:

- digital fingerprint acquisition,
- local ridge characteristic extraction,
- Ridge characteristic pattern matching.

Demonstrated model systems were evaluated based on specific performance requirements. Lockheed Martin was selected to build the AFIS segment of the FBI's

IAFIS project and the major IAFIS components were operational by 1999. Also in this timeframe, commercial fingerprint verification products began to appear for various access control, logon, and benefit verification functions [7].

An important advance in fingerprint identification was made in 1899 by Edward Henry, who established the famous “Henry system” of fingerprint classification [5,8] an elaborate method of indexing fingerprints very much tuned to facilitating the human experts performing (manual) fingerprint identification. In the early 20th century, fingerprint identification was formally accepted as a valid personal identification method by law enforcement agencies and became a standard procedure in forensics. Fingerprint identification agencies were set up worldwide and criminal fingerprint databases were established [5]. With the advent of live scan fingerprinting and availability of cheap fingerprint sensors, fingerprints are increasingly used in government and commercial applications for positive person identification. Figure 1.1 shows the fingerprint and fingerprint classification schema.



a) Arch b) Tented arch c) Right loop

d) Left loop e) Whorl e) Twin loop

Figure 1.1 Fingerprint classification schemas

Critical points in a fingerprint, called core and delta, are marked as squares and triangles. Note that an arch does not have a delta or a core. One of the two deltas in (e) and both the deltas in (f) are not imaged. A sample minutiae ridge ending (o) and ridge bifurcation (X) is illustrated in (e).

1.3 Organization of the Thesis

As a guide to the study reported in this thesis, we can make following classifications.

Chapter two gives the fundamentals of time attendance systems.

In chapter three, fingerprint classification and matching algorithms which used in this thesis will be explained.

In chapter four, general system constructions will be explained in detail.

In chapter five, some example will be given about the written program.

In chapter six, also final chapter of the thesis, conclusions of the study will be demonstrated.

CHAPTER 2

TIME ATTANDANCE & ACCESS CONTROL SYSTEMS

In this chapter of thesis we introduce some concepts called time attendance and access control systems.

2.1 Time Attendance Systems

Security and attendance systems play the most important role in the daily life. Especially, companies of all sizes use time and attendance system. Purpose of using these systems, is to gather information about employees on how much time they spend working and whether they are working regularly. The most commonly used method is the capture of an accurate time each employee reports to work and the actual hours they spend on the job. Before the invention of time tracking devices, companies recorded detailed information on each work or task of all employees performed during the day.

2.1.1 History of time attendance system

Many years ago, time tracking devices were invented to help manually recording time and attendance data on paper. IBM is the first to discover a system called International Time Recorder, located in Endicott, NY. Antique pendulum based wall clocks is the best example of systems like this and it can be found in today's markets. Its task is to manually record data to paper based cards.

Nowadays, since they are economical and easy to use small companies still use time clock systems. Easily plug the clock into the wall, set the time and start recording work hours. For systems like this, special computer skills are not required and nearly primitive computers can be used. Some manufacturers still produce these clocks, which are sold in the office products retail market. The general construction of the system is shown in Figure 2.1.

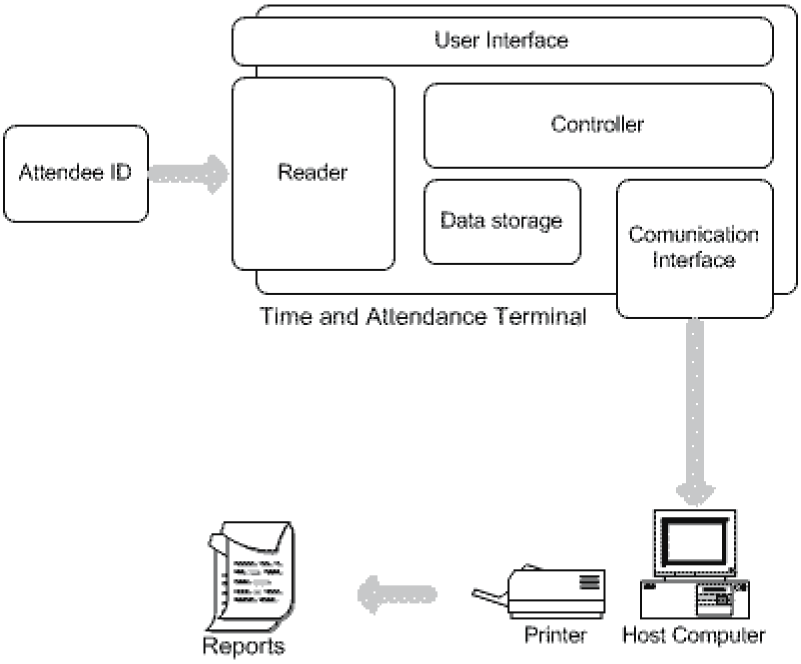


Figure 2.1 General constructions of time attendance systems

Automated Time & Attendance Systems provide an alternative to mechanical time clock systems or paper-based time sheets.

The older time and attendance system is replaced with an electronic badge reader (card reader, fingerprint reader, iris reader etc.). As soon as the employees come in the company or factory ID badge is read on the reader at the beginning and end of the work period. Then data, which are obtained by the badge reader, are stored regularly in a computer running a time and attendance software application and then the computer will automatically create timesheets for each employee.

Biometric time clocks may be alternatives for tracking time and attendance, web-based computer login stations, interactive voice response (IVR) telephony, and portable devices such as barcode scanners.

Time and attendance software play a rather important role in time and attendance systems, because all data are taken the help of this software. This software allows companies to track and evaluate the performance and work activities of employees using a single software application. At the same time, time and attendance software enables employers to store, track and organize the most important employee time related information in one place. Employee activities and management processes can be seen on a computer screen.

Companies with large number of employees usually need to install several time and attendance machines. In this case, networking becomes important RS-485 and RS-232 network protocol is commonly applied for time and attendance networking. Some systems use the more efficient networking technology, such as ethernet and WiFi.

In such system, costs vary depending on needs, with a small business system costing a few hundred dollars and larger systems supporting many complex functions costing several thousands of dollars. In addition a time and attendance system will protect a company from potential payroll fraud, provide electronic data and will give employees confidence in the accuracy of salary payments [9].

Now just before starting to explain access control systems, we review the general control system. A control system is a device or set of devices to manage, command, direct or regulate the behavior of other devices or systems. There are two common classes of control systems, with many variations and combinations: logic or sequential controls, and feedback or linear controls. There is also fuzzy logic, which attempts to combine some of the design simplicity of logic with the utility of linear control. Some devices or systems are inherently not controllable.

2.2 Open Loop Control System

An open-loop controller, also called a non-feedback controller, is a type of controller, which computes its input into a system using only the current state and its model of the system. A characteristic of the open-loop controller is that it does not use feedback to determine if its input has achieved the desired purpose. This means that the system does not observe the output of the processes. Consequently, a true open-loop system can not engage in machine learning and also cannot correct any errors that it could make. It also may not make up for disturbances in the system.

If the relationship between input and the resultant state for whichever system can be modeled by a mathematical formula an open-loop control is useful. Generally, open loop control is suitable for well-defined systems. For example determining the voltage to be fed to an electric motor that drives a constant load, in order to achieve a desired speed would be a good application of open-loop control. On the other hand, if the load were not predictable, the motor's speed might vary as a function of the load as well as of the voltage, and an open-loop controller would not be sufficient to satisfy repeatable control of the velocity.

Because of its simplicity and low-cost, an open-loop controller is often used in simple processes. Particularly, if feedback is not critical, it is used in systems. A typical example would be a conventional washing machine. The length of the machine wash time is entirely dependent on the judgment and estimation of the human. Generally, to obtain a more accurate or more adaptive control, it is necessary to feed the output of the system back to the inputs of the controller. This type of system is called a closed-loop system. Open loop control system is shown in Figure 2.2.

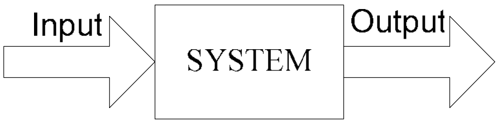


Figure 2.2 Open Loop control system

2.3 Closed Loop Control Systems

In a closed loop control system the actual error signal, which is represented between the input signal and the feedback signal by the controller in order to reduce error and get the desired value of the system's output. Feedback control systems are often referred to as closed loop controller, as shown in Figure 2.3.

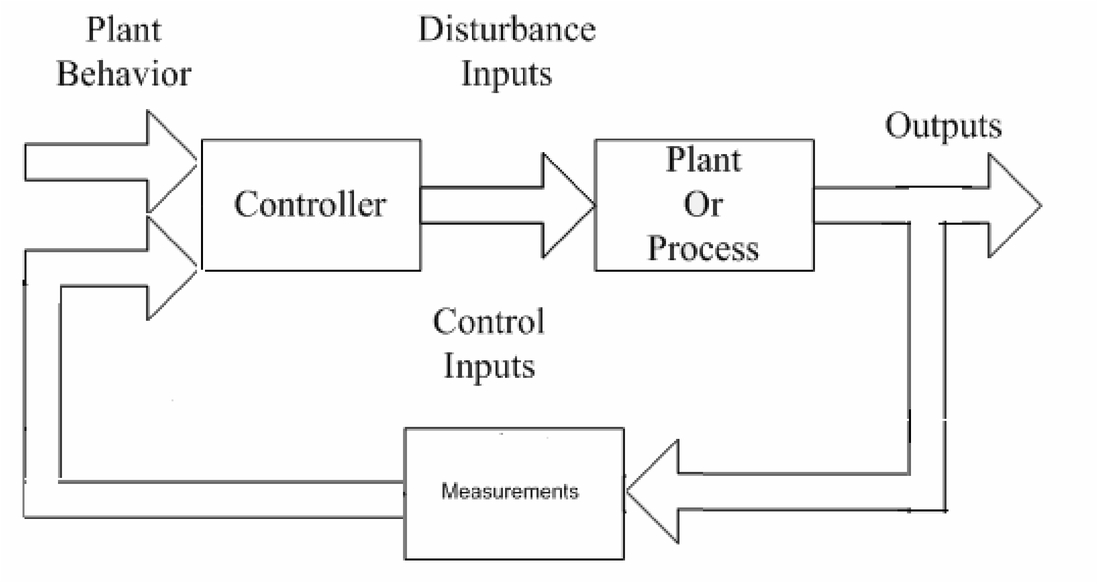


Figure 2.3 Closed loop feedback control system

The access control and time attendance system is a typical example of a closed loop feedback control system.

2.4 Access Control Systems

Nowadays, an information security system is a fundamental management responsibility. Access control includes different form of applications such as financial, privacy, safety, or defense system. In addition, access control is concerned with determining the allowed activities of legitimate users. Likewise, access control plays a role in every attempt by a user to access a resource in the system. In some systems, complete access is approved after successful authentication of the user but most systems require more complex control. In addition to the authentication mechanism (such as a password), access control is concerned with how authorizations are structured.

Generally, the purposes of an access control system are often described in terms of protecting system resources against inappropriate or undesired user access. From a business perspective, this objective can be described in terms of the optimal sharing of information. However, the main purpose of IT (information technology) is to make information available to users and applications. Access control mechanism that is equipped sufficiently can enable selective sharing of information in its absence, sharing may be considered too risky altogether [10].

Access control systems can be categorized in 4 general forms.

- Card reader access control systems
- Key Fob access control systems
- Fingerprint access control systems
- Strikes and maglocks

2.4.1 Card Reader Based Access Control Systems

Access control card readers are used in physical security systems to read a credential that allows access through access control points, typically a locked door. An access control reader can be a magnetic stripe reader, a bar code reader, a proximity reader, a smart card reader, or a biometric reader.

Access control readers may be classified by functions they are able to perform and by identification technology [11].

2.4.1.1 Bar Code

A barcode is a series of alternating dark and light stripes that are read by an optical scanner. The example of barcode and barcode reader is shown in Figure 2.4 and Figure 2.5 respectively. According to selected bar code protocol, the organization and width of the lines is changeable. In other words, the width of lines is determined by the bar code protocol. There are many different protocols. Generally, code 39 is used and the most popular in the security industry. Sometimes users can read the number of the bar code without an optical reader owing to digits, which is represented by the dark and light bars, printed bar code. Bar code technology has many advantages, since it is cheap and easy to generate the credential. Also it can easily be applied to cards or other items. The disadvantage of this technology is that it is cheap and easy to generate a credential, making the technology susceptible to fraud and the optical reader can have reliability problems with dirty or smudged credentials. One attempt to reduce fraud is to print the bar code using carbon-based ink and then cover the bar code with a dark red overlay. The bar code can then be read with an optical reader tuned to the infrared spectrum, but can not easily be copied by a copy machine. This does not address the ease with which bar code numbers can be generated from a computer using almost any printer [11].

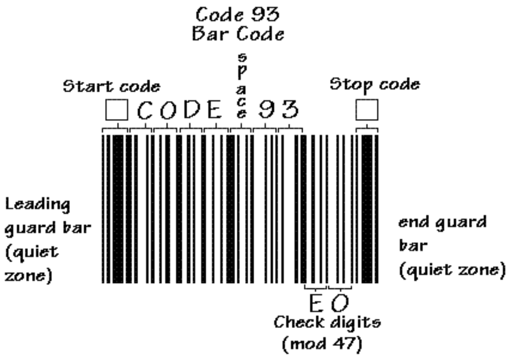


Figure 2.4 An example of a barcode

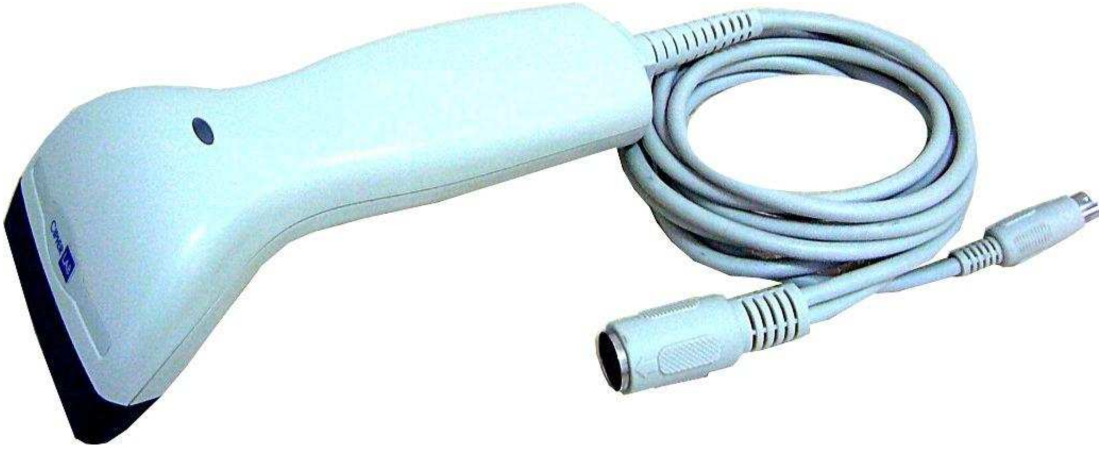


Figure 2.5 Barcode reader

2.4.1.2 Magnetic Strip

Since the stripe of magnetic oxide tape is laminated on a card the magnetic stripe technology is usually called mag-stripe. We can see a magnetic strip card in Figure 2.6. Magnetic stripe has three tracks of data. Typically the data on each of the tracks follows a specific encoding standard, but it is possible to encode any format separately. A mag-stripe card is cheap compared to other card technologies and is easy to program. The magnetic stripe holds more data than a bar code can in the same space. While a mag-stripe is more difficult to generate than a bar code, the technology for reading and encoding data on a mag-stripe is widespread and easy to acquire. Magnetic stripe technology is also susceptible to misreads, card wear, and data corruption [11].

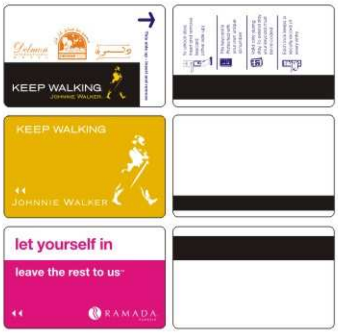


Figure 2.6 Magnetic strip cards

2.4.1.3 Wiegand Card

Wiegand card technology is a patented technology using embedded ferromagnetic wires strategically positioned to create a unique pattern that generates the identification number. Like magnetic stripe or bar code, this card which is shown in Figure 2.7 must be swiped through a reader to be read. Unlike those other technologies the identification media is embedded in the card. This technology is popular because the duplication of this card is very difficult, so the security of this card is high. Due to the limited source of supply, this technology is being replaced by proximity cards. Presently, proximity readers has the touch-less functionality and better tamper resistance, proximity cards replaced to weigand card [11].

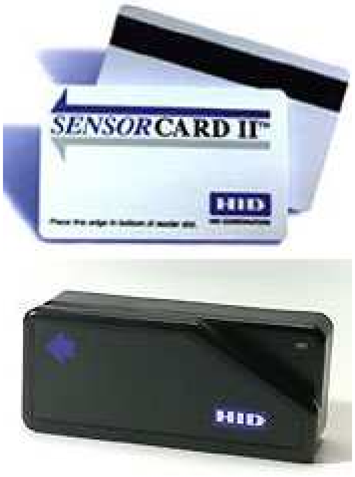


Figure 2.7 Wiegand card and its reader

2.4.1.4 Proximity Card

In the early access cards wiegand effect was used. Presently, this method is rarely used. Figure 2.8 and Figure 2.9 shows the proximity card reader and proximity access control systems respectively.

Card readers are still referred to as wiegand output readers but no longer use the wiegand effect.

The new technologies continued the wiegand upstream data so that the new readers were compatible with the old systems. A proximity reader radiates an electrical field around itself. Cards use a simple LC (Inductor-Capacitor) circuit. When a card is presented to the reader, the reader's electrical field excites a coil in the card. The coil charges a capacitor and in turn powers an integrated circuit. The integrated circuit outputs allow the person after this stage.

Figure 2.8 Proximity card reader

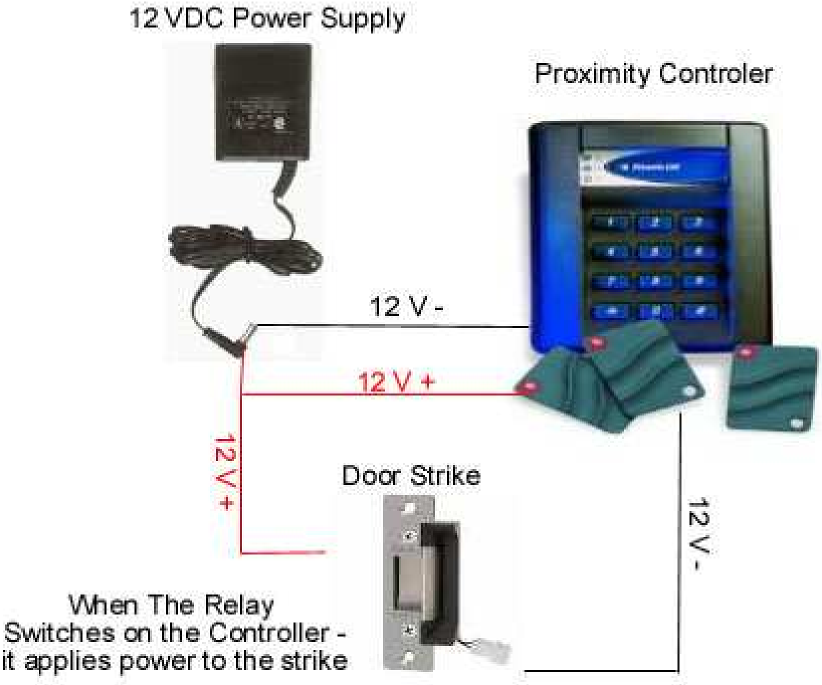


Figure 2.9 Proximity access controllers

2.4.1.5 Smart Card

There are two types of smart cards:

- Contact
- Contactless

Figure 2.10 is an example of a contact smart card. Both have an embedded microprocessor and memory. The smart card differs from the card typically called a proximity card. Smart card has microchip processor but proximity card has only microchip and task of this microchip is to provide the reader with the card's identification number. The processor on the smart card has an operating system and it can be used for different applications such as a cash card, a pre-paid membership card, and even an access control card. A contact smart card has eight contacts, which must physically touch contacts on the reader to carry information between them. Since contact cards must be inserted into readers carefully and the orientation time and speed is too long and convenience of such transaction is not acceptable for most access control applications. If the speed of transactions is not important, then such a system can be used. The working system of a contactless smart card is based on radio technology as the proximity card.

Figure 2.10 Smart card

2.4.1.6 PIN

A personal identification number (PIN) play the most important role in daily life. The PIN is usually a number consisting of four to eight digits. Less and the number is too easy to guess. More and the number is too difficult to remember. The advantage to using a PIN as an access credential is that once the number is memorized, the credential cannot be lost or left somewhere. The disadvantage is the difficulty some people have in remembering numbers that are not frequently used and the ease with which a PIN can be observed and therefore used by unauthorized people. The PIN is even less secure than a bar code or magnetic stripe card.

2.4.2 Key Fob Based Access Control Systems

The key fob is one of the forms of access control. The key fob is made from a plastic device to hang on a key ring and contains the electronics in or on it. The user holds it to the reader for access to a door. It is capable of reading and writing information to the system. This means the chip itself can be used to carry new information to a location rather than running wires throughout the building. The chip can also be placed on a company ID badge for easy access to it. Its only disadvantage is the expense of the chips [12]. Also the key fob is frequently used in cars. A use of key fob in car is shown in Figure 2.11.

Figure 2.11 An example of key fob

2.4.3 Fingerprint Based Access Control Systems

The fingerprint is probably one of the best forms of access control. Because of its uniqueness, nowadays, fingerprint is used in almost every security system or access control system. Fingerprints are one of the many forms of biometrics used to identify an individual and verify their identity. A systems like this is shown in Figure 2.12.

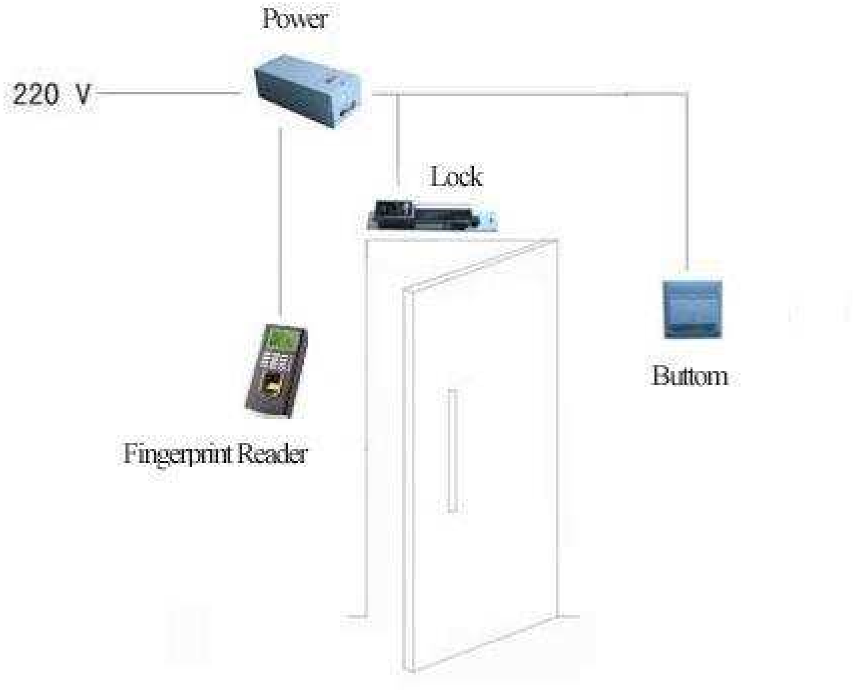


Figure 2.12 Fingerprint access control systems

2.4.4 Strikes and Maglocks

Electric strikes or magnetic locks can be used to further regulate who can pass through access points, providing an even greater level of protection.

Electric strikes are door locking devices, usually solenoid-operated that will unlock the door when electrical power is applied to it. An opening that requires a person to be “buzzed in” is equipped with an electric strike. The buzzing sound is created when a button is pushed, sending an AC current through the device. This action disengages the device and allows the door to open. The operation just mentioned is a fail secure mode of operation, the most common function of an electric strike. A fail safe configuration will operate in the reverse condition; normally locked when power is applied and unlocked when power is interrupted. If desired, the buzzing sound can be eliminated by using a DC power source [12]. A simple example of maglock is shown in Figure 2.13.

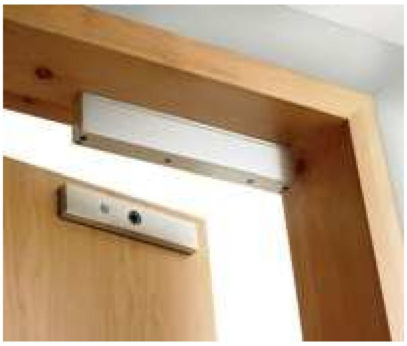


Figure 2.13 Maglock

CHAPTER 3

FINGERPRINT CLASSIFICATION AND MATCHING

3.1 General System Definitions

Fingerprint recognition algorithm that identifies the person can be classified into two essentially distinct type of issue with different category [1]. First one is recognition and the second one is verification.

- **Recognition** is responsible for the issue of establishing a subject's identity
- **Verification** is responsible for the issue of accepting or rejecting a person claiming entrance.

For example, computer technology has an important role on the access controls that is based on the human's physical features. Using these properties, human identity theft can be prevented. Typically, a person could be identified based on verification of a person's belongings, such as:

- Physical access to a building by an authenticated key
- Login access to a system with a user ID and associated password that is original to person

Another method is based on identifying physical characteristics of the person. The characteristics could be either a person's physiological features, such as fingerprints, hand geometry, etc. or behavioral characteristics, such as voice and signature. This method of identification of a person based on behavioral characteristics is called biometrics. Since the biological characteristics can not be forgotten (like passwords) and can not be easily shared or misplaced (like keys), they are generally considered to be a more reliable approach to solving the personal identification problem.

3.2 Fingerprint as a Biometric

Fingerprints are one of the most well known biometric technologies and are considered reliable proofs of evidence in courts of law all around the world. Therefore, they are used in forensic divisions worldwide for criminal investigations. Fingerprint has complex structure. This structure is shown in Figure 3.1. More recently, most of commercial applications are either using or considering to use fingerprint-based identification systems due to demonstration of superior performance than any other existing biometric authentication systems.

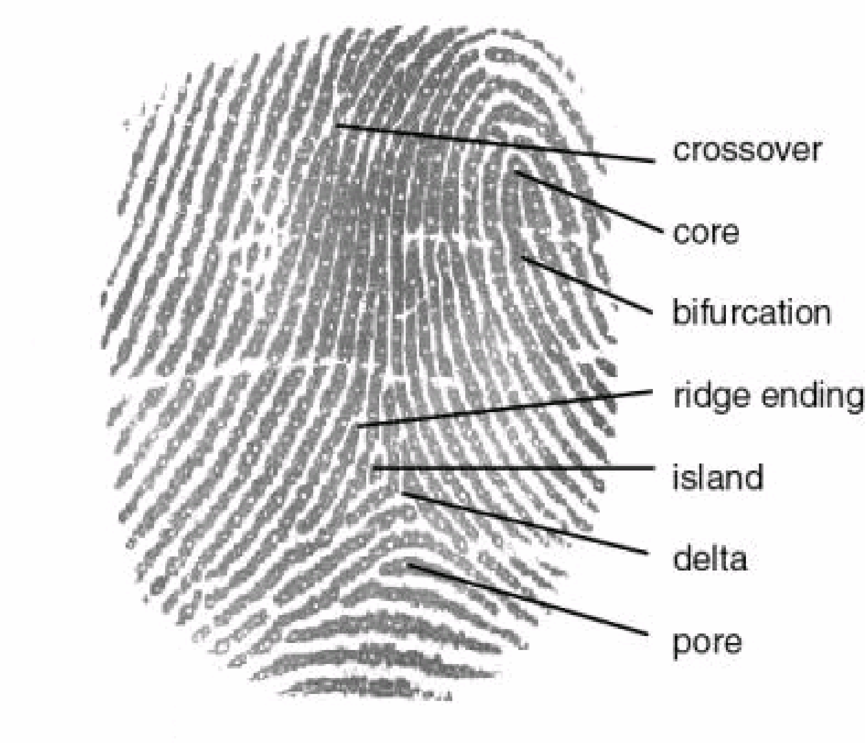


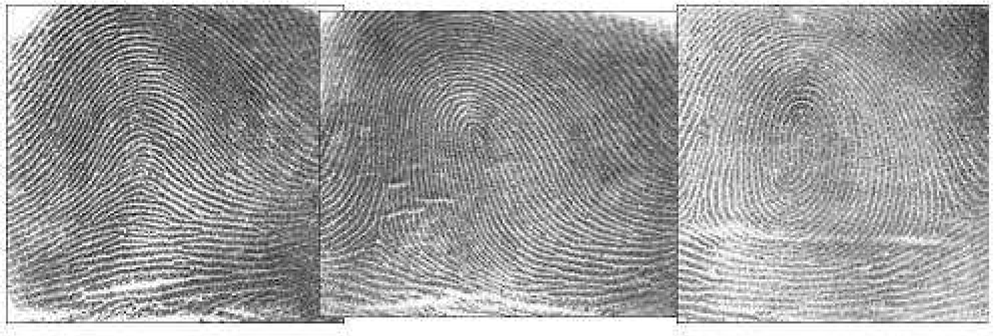
Figure 3.1 Structure of fingerprint

3.3 General System Architecture

In order to analyze fingerprint for matching purposes, we need to compare several features of the print pattern. These features are unique patterns, which combine characteristics of ridges, and minutia points, which are given in Figure 3.2. In order to successfully employ some of the imaging technologies, we also need the structure and properties of human skin.

3.3.1 Patterns

Patterns of fingerprint ridges compose of three parts called, the arch, loop, and whorl. These patterns are shown in Figure 3.2 respectively. From the Figure 3.2 a), ridge entering from one side of the finger and rising in the center forming an arc and exiting the other side of the finger is called the arch. Similarly, ridge entering from one side of the finger, forming a curve and exiting from the same side is called the loop. If the ridges are formed in a circular shape around the central point, it is called the whorl. Arch, loop, whorl patterns are shown in Figure 3.2.



a) The arch pattern b) The loop pattern c) The whorl pattern

Figure 3.2 Different finger patterns

3.3.2 Properties of Minutia

The major minutia features of fingerprint ridges consist of three parts.

- ridge ending,
- bifurcation,
- short ridge (or dot)

The point where ridge terminates is called ridge ending. If a single ridge splits into two ridges, they are called bifurcations. Short ridges (or dots) are ridges, which are significantly shorter than the average ridge length on the fingerprint. Minutiae and patterns are very important in the analysis of fingerprints since no two fingers have been shown to be identical**.**

Minutia features consist of three basic concepts as shown in Figure 3.3. These are ridge ending, bifurcation, short ridge.

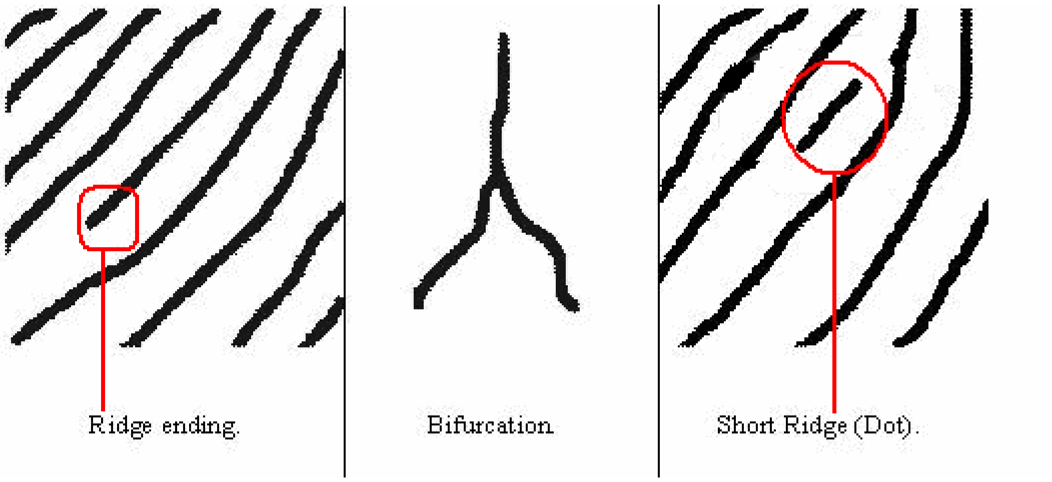


Figure 3.3 Concepts of minutia

3.3.3 System Architecture

Fingerprint based automatic identity authentication system's general architecture consists of four components. These are :

- User interface
- System database
- Registration (enrollment) module
- Authentication module

Firstly, to enter the system, user fingerprints are introduced to the system. The **user interface** provides this. The function of user interface is to provide mechanisms for a user to indicate his/her identity and input his/her fingerprints into the system. The **system database** consists of a collection of records which corresponds to an authorized person that has access to the system. Records contain some information, which belongs to the person.

- ➢ User name
- ➢ Minutiae templates
- ➢ Other information

The **registration module** records persons and their fingerprints into the system database. If the fingerprint images of a person are recorded in the enrollment module, a minutiae extraction algorithm is applied to the fingerprint images. Then minutiae patterns are extracted. Using the records in the system database only consists of fingerprints of good quality, a quality checking algorithm is ensured. If a fingerprint image is of insufficient quality, clarity of ridge and valley structures all the regions cannot be reliably recovered. The procedure goes on and enhanced fingerprint image is fed to the minutiae extractor again. This procedure is shown in Figure 3.4

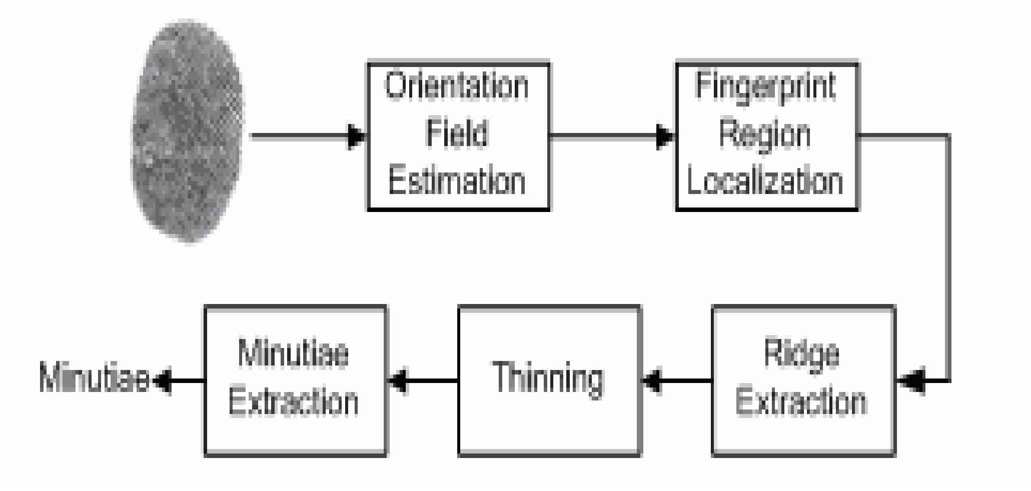


Figure 3.4 How to obtain minutia

The task of **authentication module** is to authenticate the identity of the person who desires to access the system. The person indicates their identity and is authenticated by the authentication module, then places his/her finger on the fingerprint scanner and a digital image of his/her fingerprint is captured. Minutiae pattern is extracted from the captured fingerprint image and fed to a matching algorithm which matches it against the person’s minutiae templates stored in the system database to establish the identity [12]. Fingerprint based automatic identity authentication system’s general construction is shown in Figure 3.5.

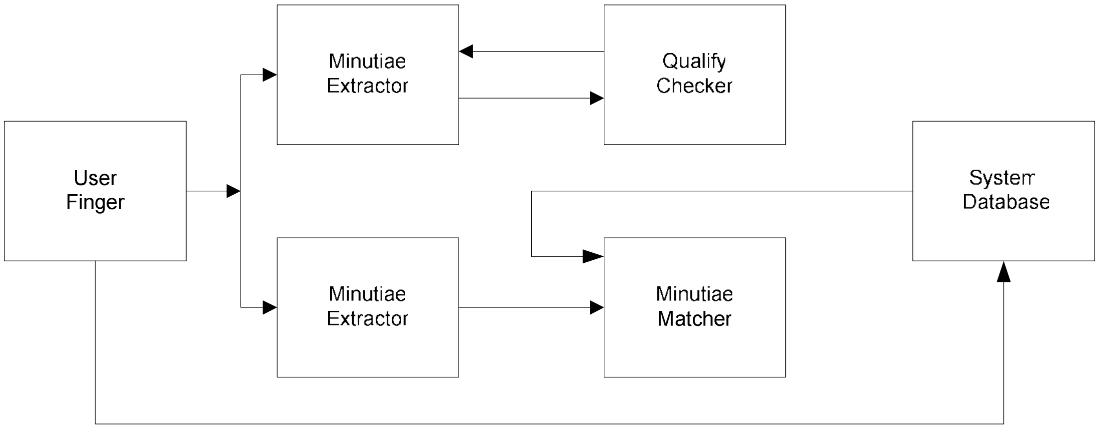


Figure 3.5 Architecture of an automatic identity authentication system

3.4 Fingerprint Sensing

Fingerprint images can be captured with two primary methods. These methods are:

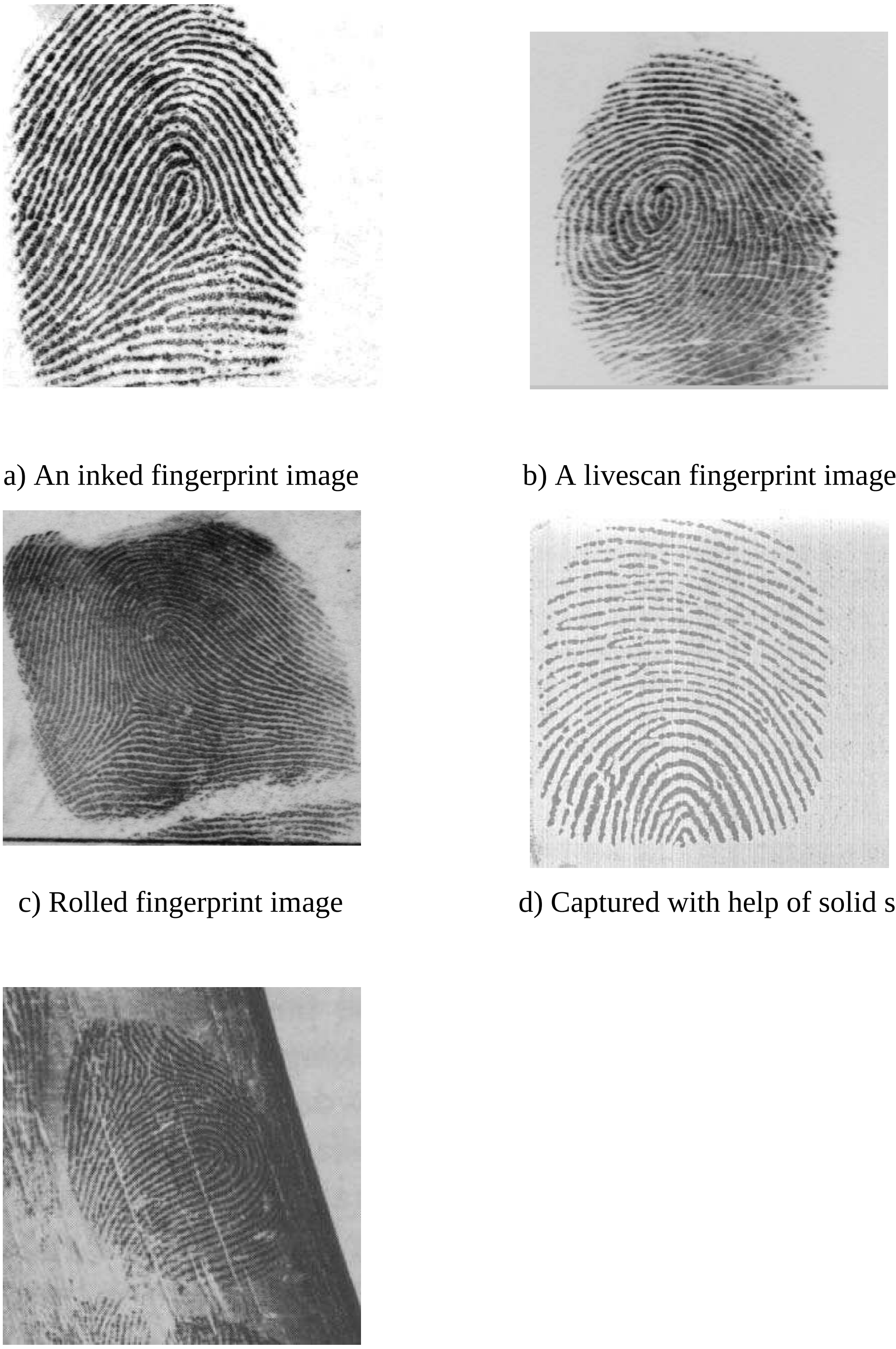
- inked (off-line)
- live scan (ink-less)

An inked fingerprint image is typically obtained in the following way.

An **inked** finger is placed on a paper and pressed. Then obtained image is scanned using a scanner.

Nevertheless, the **live scan** fingerprint is obtained directly from the finger. This method is used in the following way:

Fingerprint image is directly obtained from the finger, a collective term must be obtained only, without the intermediate step of getting an impression on a paper. To obtained the inked fingerprints is unnecessary; in the context of an identity authentication system, it is both negative and unacceptable. Nowadays, to obtain a live-scan fingerprint image, the most popular technology is based on optical frustrated total internal reflection (FTIR) concept [14]. Figure 3.6 shows different fingerprint images.



a) An inked fingerprint image

b) A livescan fingerprint image

c) Rolled fingerprint image

d) Captured with help of solid state sensor

e) A latent fingerprint image

Figure 3.6 Fingerprint sensing

If the images shown in figure are explained briefly;

- *(a) An inked fingerprint image could be captured from the inked impression of a finger;*
- *(b) A livescan fingerprint is directly imaged from a live finger based on optical total internal reflection principle: the light scatters where finger (e.g., ridges) touch the glass prism and light reflects where finger (e.g., valleys) does not touch the glass prism.*
- *(c) Rolled fingerprints are images depicting nail-to-nail area of a finger*
- *(d) Fingerprints captured using solid state sensors show a smaller area of finger than a typical fingerprint dab captured using optical scanners.*
- *(e) A latent fingerprint refers to partial print typically lifted from a scene of crime.*

Running principle of **frustrated total internal reflection** is in following way:

When a finger is placed on one side of a glass platen (prism), ridges of the finger are in contact with the platen, while the valleys of the finger are not in contact with the platen. The rest of the imaging system essentially consists of an assembly of an LED light source and a CCD placed on the other side of the glass platen. The laser light source illuminates the glass at a certain angle and the camera is placed such that it can capture the laser light reflected from the glass. The light incidenting on the platen at the glass surface touched by the ridges is randomly scattered while the light incidenting at the glass surface corresponding to valleys suffers total internal reflection. Consequently, portions of the image formed on the imaging plane of the CCD corresponding to ridges are dark and those corresponding to valleys are bright. More recently, capacitance-based solid state live-scan fingerprint sensors are gaining popularity since they are very small in size and hold promise of becoming inexpensive in the near future. A capacitance-based fingerprint sensor essentially consists of an array of electrodes. The fingerprint skin acts as the other electrode, thereby, forming a miniature capacitor. The capacitance due to the ridges is higher than those formed by valleys [13]. This differential capacitance is the basis of operation of a capacitance-based solid state sensor [15].

3.5 Fingerprint Sensor

A fingerprint sensor is an electronic device. Fingerprint sensor is used to capture a digital image of the fingerprint pattern. The captured image is called a live scan. To create a biometric template, this live scan is digitally processed (a collection of extracted features) which is stored and used for matching. Fingerprint sensors can be categorized in the following way.

3.5.1 Optical

Optical fingerprint imaging involves capturing a digital image of the print using visible light. This type of sensor is especially used with a specialized digital camera. The top layer of the sensor, where the finger is placed, is known as the touch surface. Under this layer is a light-emitting phosphor layer, which illuminates the surface of the finger. The light reflected from the finger passes through the phosphor layer to an array of solid-state pixels (a charge-coupled device) which captures a visual image of the fingerprint. A dirty or scratched touch surface can cause a bad image of the fingerprint. The most disadvantage of this type of sensor is the fact that the imaging capabilities are affected by the quality of skin on the finger. For instance, a dirty or marked finger may be difficult to image properly. In addition, it is possible for an individual to erode the outer layer of skin on the fingertips to the point where the fingerprint is no longer visible. If not coupled with a "live finger" detector, it can also be easily fooled by an image of a fingerprint. However, unlike capacitive sensors, this sensor technology is not susceptible to electrostatic discharge damage.

3.5.2 Ultrasonic

In order to create visual images of the fingerprint, ultrasonic sensors make use of the principles of medical ultrasonography. An ultrasonic sensor uses very high frequency sound waves to penetrate the epidermal layer of skin, unlike optical imaging. The sound waves are generated using piezoelectric transducers. In addition, to measure the reflected energy, piezoelectric materials are used. The reflected wave measurements can be used to form an image of the fingerprint because the dermal skin layer exhibits the same characteristic pattern of the fingerprint. This eliminates the need for clean, undamaged epidermal skin and a clean sensing surface.

3.5.3 Capacitance

In order to form fingerprint images, capacitance sensors use the principles associated with capacitance. The most commonly known two equations used in this type of imaging are:

$$C = \frac{Q}{V} \ , \ C = \in_o \in_r \frac{A}{d}$$

Where;

- C is the capacitance in farads
- Q is the charge in coulombs
- V is the potential in volts
- $\in_o$ is the permittivity of free space, measured in farad per meter
- $\in_r$ is the dielectric constant of the insulator used
- A is the area of each plane electrode, measured in square meters
- d is the separation between the electrodes, measured in meters

In this method, to obtain imaging, the sensor array pixels each act as one plate of a parallel-plate capacitor. The dermal layer (which is electrically conductive) acts as the other plate, and the non-conductive epidermal layer acts as a dielectric. Capacitance sensors can be evaluated as two main categories. First one is passive and second one is active. In the following part, necessary information about the capacitance sensors is given.

3.5.3.1 Passive Capacitance

To form an image of the fingerprint patterns on the dermal layer of skin, a passive capacitance sensor uses the principle summarized above. Here, each sensor pixel is used to measure the capacitance at that point of the array. The capacitance varies between the ridges and valleys of the fingerprint, because the volume between the dermal layer and sensing element in valleys contains an air gap. The dielectric constant of the epidermis and the area of the sensing element are known values. The measured capacitance values are then used to distinguish between fingerprint ridges and valleys.

3.5.3.2 Active Capacitance

Active capacitance sensors use a charging cycle to apply a voltage to the skin before measurement occurs. The application of voltage charges the effective capacitor. The electric field between the finger and sensor follows the pattern of the ridges in the dermal skin layer. On the discharge cycle, the voltage across the dermal layer and sensing element is compared against a reference voltage in order to calculate the capacitance. Mathematically, the distance values are then calculated using the above equations, and used to form an image of the fingerprint. Active capacitance sensors measure the ridge patterns of the dermal layer like the ultrasonic method. In addition, this eliminates the need for clean, undamaged epidermal skin and a clean sensing surface.

3.6 Fingerprint Representation

Fingerprint representations are shown with two different types:

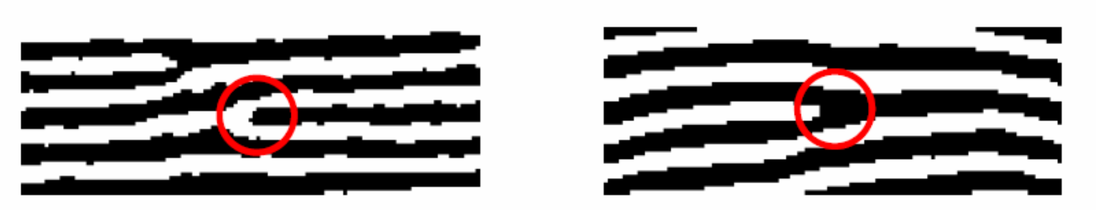
- Local
- Global

Major representations of the local information in fingerprints are based on the entire image, finger ridges, pores on the ridges, or outstanding features derived from the ridges. Representations normally based on ridge endings or bifurcations, which are shown in Figure 3.7 (compose of ridge ending and bifurcation known as minutiae), are the most common, primarily due to the following reasons.

- Minutiae capture much of the individual information
- Minutiae-based representations are storage-efficient

- Minutiae detection is relatively robust to various sources of fingerprint collapse.

Minutiae-based representations, generally, rely on locations of the minutiae and the directions of ridges at the minutiae location. In other words, some global representations include information about locations of critical points (e.g., core and delta) in a fingerprint.



a) Ridge ending

b) Ridge bifurcation

Figure 3.7 Ridge ending and ridge bifurcation

3.7 Feature Extraction

Given the input fingerprint images, a feature extractor finds the ridge endings and ridge bifurcations. If ridges can be smoothly located in an input fingerprint image, then minutiae extraction is extracting singular points in a thinned ridge map. In practice, nevertheless, it is not possible, generally, to obtain a perfect ridge map. Clearly, the performance of currently available minutiae extraction algorithms depends on the quality of the input fingerprint images. Especially, many factors (aberrant formations of epidermal ridges of fingerprints, postnatal marks, occupational marks, problems with acquisition devices, etc.), fingerprint images may not always have well-defined ridge structures.

Performance of the automatic identity authentication system which uses fingerprints depends on a reliable minutiae extraction algorithm. This is quite important and critical.

The overall flowchart of a typical minutia-extracting algorithm that is indicated above is shown in Figure 3.8. It mainly consists of three components:

- Orientation field estimation
- Ridge extraction
- Minutiae extraction and post processing

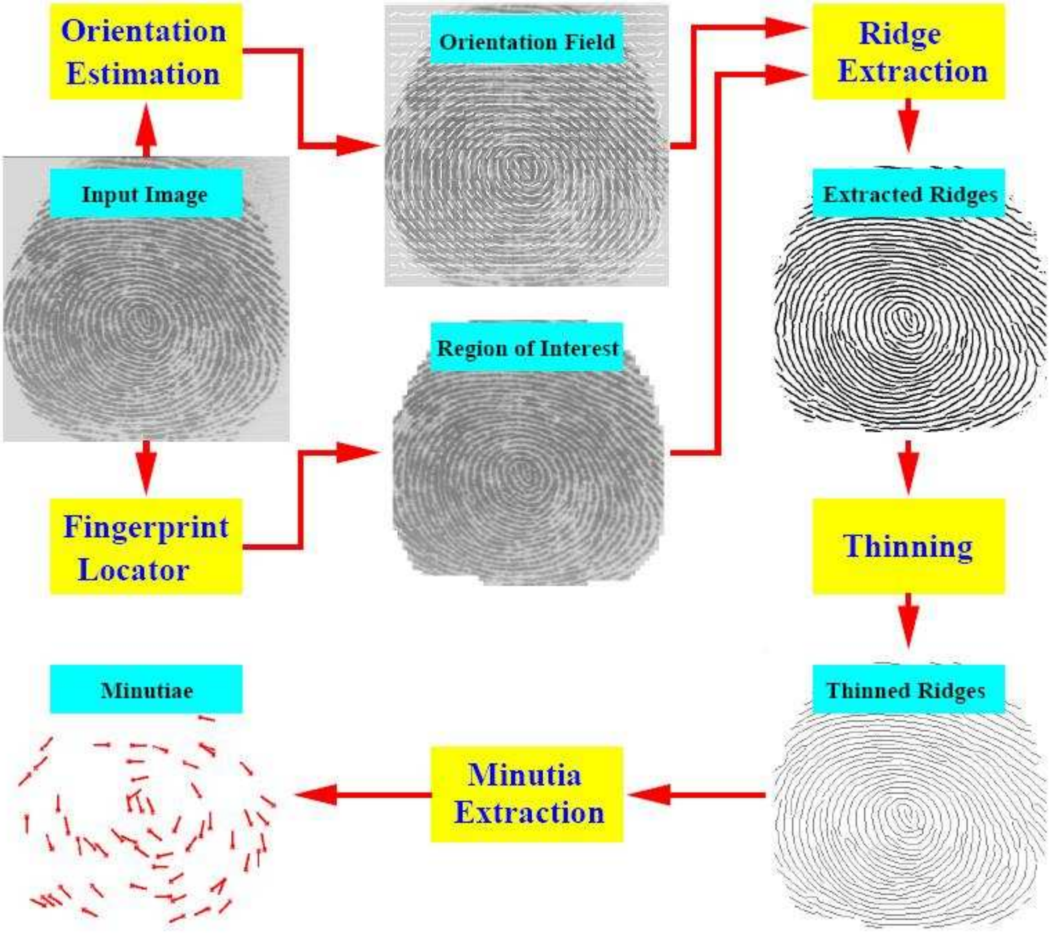


Figure 3.8 Flowchart of the minutiae extraction algorithm [13]

3.7.1 Orientation Estimation

The directionality of ridges in the fingerprint image is represented by the orientation field of a fingerprint image. It plays a very important role in fingerprint image analysis. Up to now, many methods have been suggested to evaluate the orientation field of fingerprint images [14]. Typically, fingerprint image is divided into a number of non-overlapping blocks. Moreover, an orientation representative of the ridges in the block is allocated to the block based on an analysis of grayscale gradients in the block. The block orientation could be determined from the pixel gradient orientations based on averaging [14], voting [16], or optimization [17].

3.7.2 Segmentation

The most important problem of the minutia extracting is to localize the parts of fingerprint image describing the finger foreground. In the simplest of approaches is to segment the foreground by global or adaptive thresholding. Up to now, a novel and reliable approach to segmentation by (Ratha et al.) [20] is an important difference in the magnitudes of variance in the graylevels along and across the flow of a fingerprint ridge.

3.7.3 Ridge Detection

Ridge detection of the approaches is using either simple or adaptive thresholding. However, the most important fault of these approaches is that these may not work for noisy and low contrast parts of the image. The other important property of the ridges in a fingerprint image is that the gray level values on ridges obtain their local maxima along a direction normal to the local ridge orientation [13, 17]. Pixels can be identified to be ridge pixels based on this property. The extracted ridges may be thinned/cleaned using standard thinning and connected component algorithms.

3.7.4 Minutiae Detection

As soon as the thinned ridge map is available, the ridge pixels with three ridge pixel neighbors are identified as ridge bifurcations. Then identified ridge bifurcation with one ridge pixel neighbor is identified as ridge endings. Nevertheless, all the minutia, which are detected in this way, are not true because of the noise in the fingerprint image.

3.7.5 Post-processing

Generally, at this stage, genuine (true) minutiae are obtained from the extracted minutiae using a number of techniques. For example, too many minutiae in a small neighborhood may indicate noise and they could be discarded. Sometimes, very close ridge endings oriented anti-parallel to each other may indicate affected (false) minutia. This problem, generally, is generated due to either poor contrast or an injury in the finger. Often, two very closely located bifurcations sharing a common short ridge suggest foreign minutia generated by bridging of neighbor ridges because of dirt or image processing artifacts.

3.8 Fingerprint Enhancement

If the input fingerprint image quality is very high, then the performance of a fingerprint image matching algorithm is very good because the quality depend on input fingerprint images. In practice, a significant percentage of acquired fingerprint images are poor quality. In poor-quality fingerprint images, ridge structures are not always well-defined and so they can not be correctly detected. This causes the following problems:

i) Significant number of foreign minutiae may be created
ii) A large percentage of true minutiae may be ignored
iii) Large errors in minutiae localization may be introduced.

If an enhancement algorithm that can improve the clarity of the ridge structures exists, then the performance of the minutiae extraction algorithm will be strong with respect to the quality of fingerprint images.

The ridges and creases, in a small local neighborhood, approximately form a two-dimensional sinusoidal wave along an orthogonal direction to local ridge orientation. Similarly, the ridges and creases in a small local neighborhood have well-defined local frequency and local orientation properties. According to scientists, common approaches employ a band pass filter that models the frequency domain characteristics of a good quality fingerprint image. The poor quality fingerprint image is processed using the filter to block and avoid the foreign noise and pass the fingerprint signal. Some methods may estimate the orientation or frequency of ridge in each block in the fingerprint image and adaptively tune the filter characteristics to match the ridge characteristics.

One typical variation of this theme segments the image into non-overlapping square blocks of widths larger than the average inter-ridge distance. Using a bank of directional band pass filters, each filter is matched to a predetermined model of generic fingerprint ridges flowing in a certain direction; the filter generating a strong response indicates the dominant direction of the ridge flow in the finger in the given block. The resulting orientation information is more accurate, leading to more reliable features. A single block direction can never truly represent the directions of the ridges in the block and may consequently introduce filter artifacts.
For instance, one common directional filter used for fingerprint enhancement is a Gabor filter. Gabor filters have both frequency-selective and orientation-selective properties and have optimal joint resolution in both spatial and frequency domains. The even-symmetric Gabor filter has the general form :

$$h(x,y)=\exp\left\{-\frac{1}{2}\left[\frac{x^2}{\delta_x^2}+\frac{y^2}{\delta_y^2}\right]\right\}\cos(2\pi u_0 x)$$

Where u_0 *is the frequency of a sinusoidal plane wave along the x-axis, and* δ_x *and* δ_y *are the space constants of the Gaussian envelope along x and y axes, respectively. Gabor filters with arbitrary orientation can be obtained via a rotation of the x-y coordinate system* [13].

Up to now, we have explained how to obtain the minutia patterns or how to filter image. Now, we can see in Figure 3.9 and Figure 3.10 the fingerprint enhancement algorithms and fingerprint enhancements results respectively.

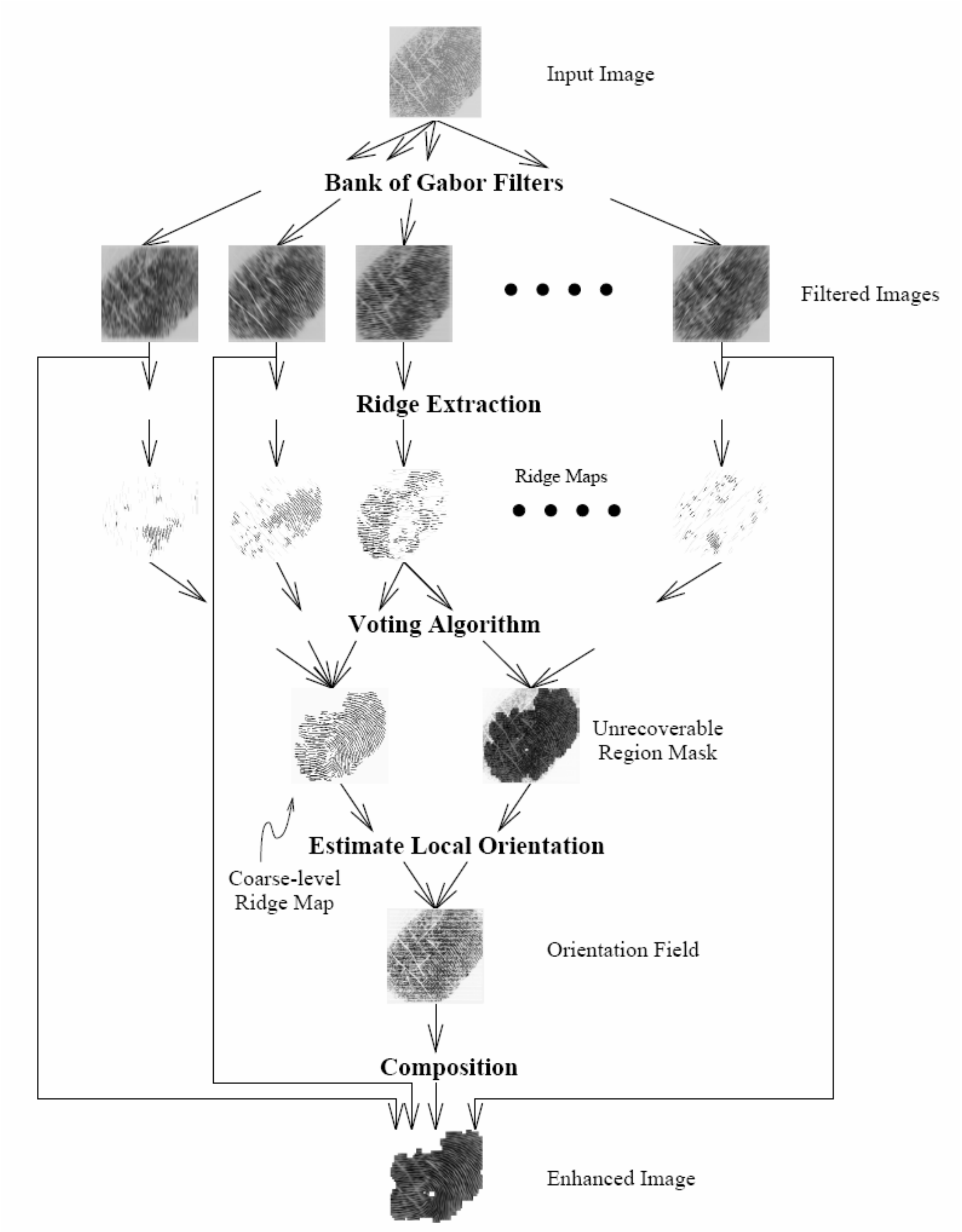
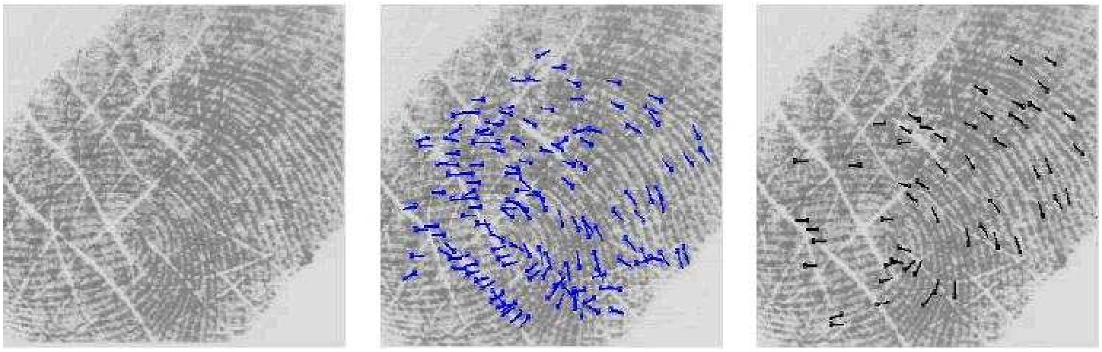


Figure 3.9 Fingerprint enhancement algorithms [13]



a) A poor quality fingerprint **b)** Minutia extracted without image enhancement **c)** Minutiae extracted after image enhancement

Figure 3.10 Fingerprint enhancement results

CHAPTER 4

DESIGN AND CONSTURUCTION OF FINGERPRINT SYSTEM

Fingerprint based door lock system generally have four common elements. These are: fingerprint device, interface (hardware), software and door lock. In this chapter, process of the system will be introduced and also fingerprint device, hardware and door lock used in this study will be introduced in order to provide some brief information about their working principles. Figure 4.1 shows a schematic diagram of the system.

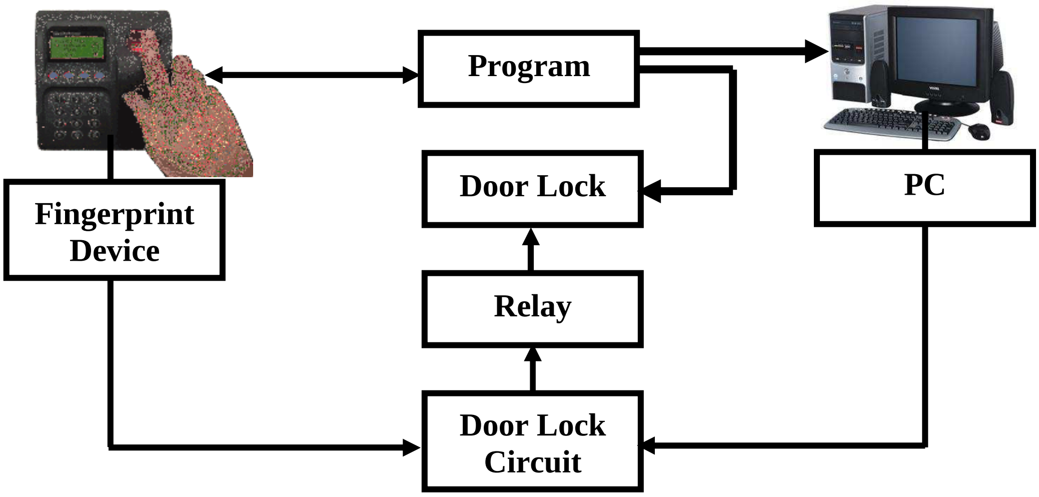


Figure 4.1 A schematic diagram of the system

When a finger is placed on the fingerprint device, the device checks the person whether authorized. If the person is authorized, the device sends data to program interface to communicate between the device and PC. The communication between PC and device can be established by TCP/IP port or serial port such as RS232 or RS485 port. However, generally, TCP/IP port is preferred than serial ports because TCP/IP is more suitable and rather fast compared to serial. After the communication is satisfied, PC controls the person to whether open the door because the person may not be authorized to pass this door. If the person is authorized for this door, the door is opened by the help of written program. So the process is done.

The most important advantage of the system is that the connection is made by TCP/IP and owing to this connection the program is set-up and run any computer which connected on the same network and all progress can observed and reported belong to any employees. Also, to allow the guests entering an icon is placed on the program menu. This feature is independent the device and controlled by program user manually. Reports can be taken by select of date range. This feature provides high performance.

4.1 Fingerprint Device

Fingerprint device is used to retrieve finger image and matching. This device is composed of some unit such as digital information screen, keypad, fingerprint sensor, RJ 45 jack connector, RS 232 or RS 485 serial port and warning light. In this study, a X628 fingerprint T&A device is used.

This device is a fairly developed model. The most important feature of this device is that other devices scan 4 points of 8 points to match, but this device scans 4 points of 24 points. This provides high performance on injured, deformed, or scratched fingers.

The fingerprint device [18] used in this study is shown in Figure 4.2 and Table 4.1 shows its technical specifications.

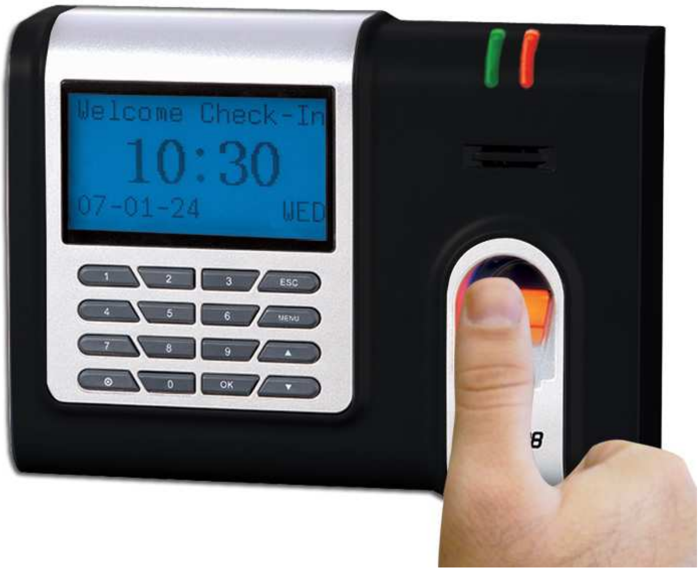


Figure 4.2 Fingerprint device

Table 4.1 Technical specification of fingerprint device

Technical Specifications
· User Capacity_ 2200/2800 · Transaction Storage _ 80000/120000 · Algorithm version: Biokey VX6.0 · Hardware platform: ZEM100 · Communications: RS232, RS485&TCP/IP · FAR: ≤0.001% · FRR: <= 1% · Operating Temperature: 0- 45 · Operating Humidity: 20% - 80% · Sensor: ZK Sensor · 26-bit Weigand In/Out · Size:190(L)*140(w)*57(H) · Language: Simplified Chinese Traditional Chinese English and other languages

4.1.1 Hardware platform (ZEM100)

Fingerprint embedded standalone module (ZEM100) is built in strong RISC processor and excellent, reliable, high-speed fingerprint algorithm; it is easy to be integrated with various systems. The average identification speed is less than 3 seconds for 2000 pieces fingerprint template; 2M Flash on board can storage 2,800 pieces fingerprint template and 120,000 transaction records, 4500 pieces fingerprint template and 300,000 transaction records is available for customization. The ZEM100 includes a fingerprint identification module which is smaller than ID card and a reliable optical fingerprint sensor, suitable for various applications (door lock, access control etc.) [19].

Figure 4.3 shows the ZEM 100 and Table 4.2 shows its technical specifications.

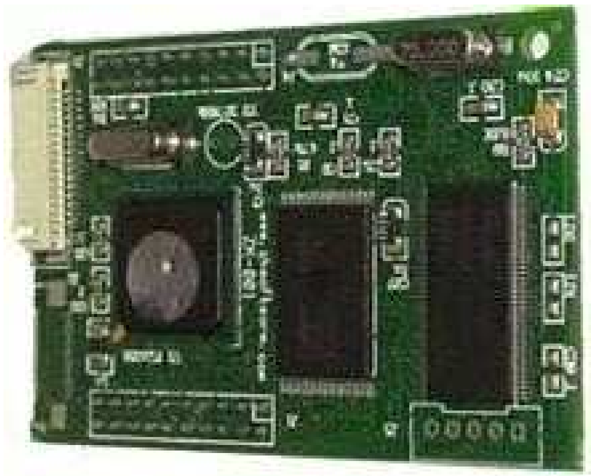


Figure 4.3 ZEM 100

Table 4.2 Technical Specifications of ZEM 100

Technical Specifications
· Size (Optical sensor)：60*25*23mm
· Size (Board)：71*45*8mm
· Operating Temperature： -20° - 55°C
· Operating Humidity： < 85%
· Storage Temperature： -40° - 80°C
· Image Resolution： 500DPI
· Effective Sensing Area： 15*15mm
· Image Grayscale： 256 Shades(8bit gray level)
· 1:N Matching Speed： 3s (2000 pieces, Average)
· FRR： 1.4%
· FAR： 0.00001%
· Supply voltage： 5V±5%
· Power： <1.5W
· Wiegand：Output Standard 26Bit/CustomFormats
· Communication Interface(Base ZKAPI protocol) ：Standard Serial interface Half-duplex mode Serial Interface Ethernet interface (optional)
· User capacity：2800
· Transaction capacity：120000

4.2 Door Lock System

The designed system consists of electronic components. Main function of this hardware is to open the door lock. Working principle of this system can summarize as following:

As soon as an authorized person scans her/his finger to the fingerprint device the device generates a very small voltage. Generated voltage is approximately 0,2 V. This voltage is carried by the help of Cat5 cable to door lock circuit. But this voltage is not sufficient to run the system. The voltage must be increased. By the help of an amplifier, the generated voltage is increased level of 0,7 V. Thus, the input voltage of transistor is satisfied. To feed of amplifier +Vcc and –Vcc a transformer is used. The transformer turns ratio is 220/12. Namely, 12V is obtained in the secondary part of the transformer. Afterwards, to obtain rectified positive and negative voltage two diodes are uses. Two capacitors are placed on the diode end line to filter the incoming voltage. LM7812 and LM7912 are uses to obtain smooth and stable voltage. LM7812 generates positive 12 volt and LM7912 generates negative 12 volt. These voltages are uses to feed amplifier. After the amplifier is fed, amplifier generates a sufficient voltage to feed the transistor. The voltage is sent to base of transistor and transistor is going to on position. Thus, relay is going from the normally close position to normally open position. By the help of connectors which is placed on the door lock system box, output is satisfied and the door is opened and authorized person can enter into the desired place. The designed door lock system is shown in the Figure 4.4.

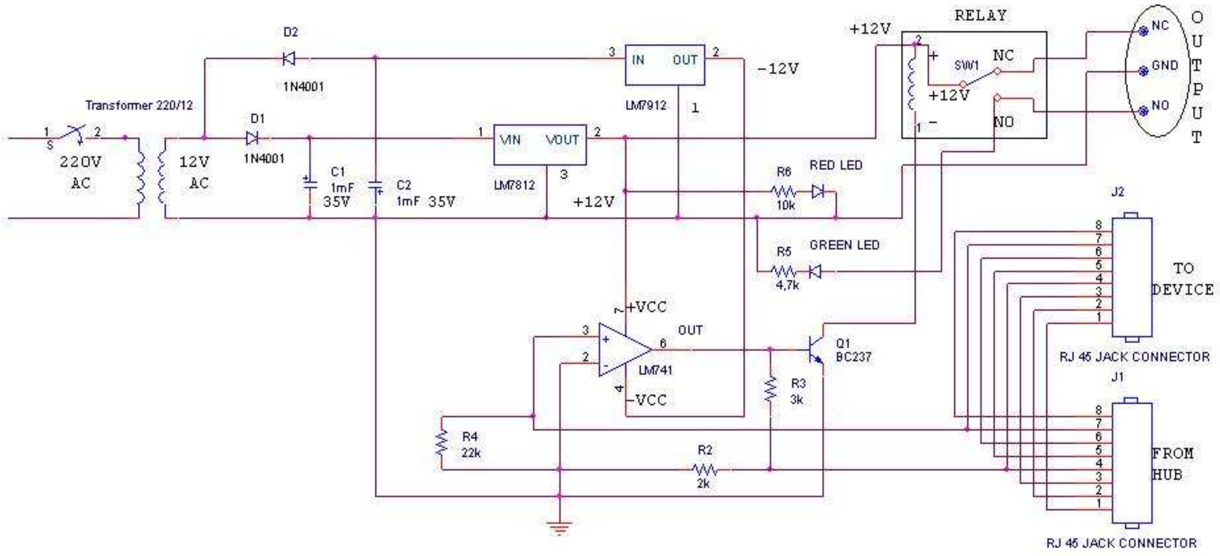


Figure 4.4 The designed door lock opening system

Now, the door lock system is briefly explained.

4.2.1 Diode bridge

A diode bridge or bridge rectifier is an arrangement of four diodes in a bridge configuration that provides the same polarity of output voltage for either polarity or input voltage. When used in its most common application, for conversion of alternating current (AC) input into direct current (DC) output, it is known as a bridge rectifier [20].

In this study, the diode bridge is used in order to obtain DC voltage. Figure 4.5 shows the diode bridge samples.

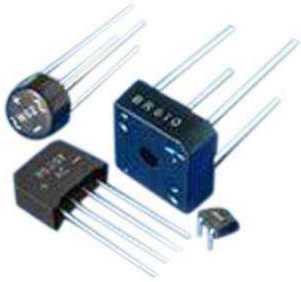


Figure 4.5 Diode bridge

4.2.2 Operational amplifier (LM741)

In this study, LM741 is used to increase incoming voltage that to the desired level. If an authorized person scans her/his finger to the device the device generates a voltage. But the voltage is not the desired voltage and in order to increase this voltage, we must use the amplifier. LM 741 can easily do this work. LM 741 has 8 pins and Figure 4.6 shows its composition and Figure 4.7 shows its connection diagrams.

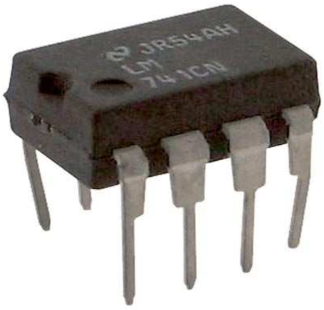


Figure 4.6 LM 741

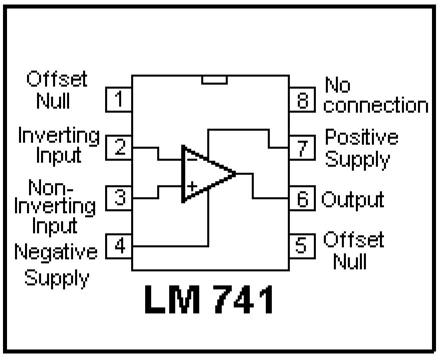


Figure 4.7 Connection diagram of LM 741

4.2.3 BC 237 transistor

BC 237 is used to switching. Table 4.3 shows the parameters of BC237.

Table 4.3 Parameters of BC 237 Transistor

SYMBOL PARAMETER CONDITIONS MIN. MAX. UNIT
VCBO collector-base voltage open emitter – 50 V
VCEO collector-emitter voltage open base – 45 V
VEBO emitter-base voltage open collector – 6 V
IC collector current (DC) – 100 mA
ICM peak collector current – 200 mA
IBM peak base current – 200 mA
Ptot total power dissipation Tamb ≤ 25 °C; note 1 – 500 mW
Tstg storage temperature –65 +150 °C
Tj junction temperature – 150 °C
Tamb operating ambient temperature –65 +150 °C

CHAPTER 5

PROGRAMMING OF ACCESS CONTROL SYSTEM

The main system construction is given in the 4th chapter. This chapter is related to the designed system program. The program consists of 3 main sections. These are;

- Device definitions
- Department definitions
- Personal definitions

Figure 5.1 shows these sections.

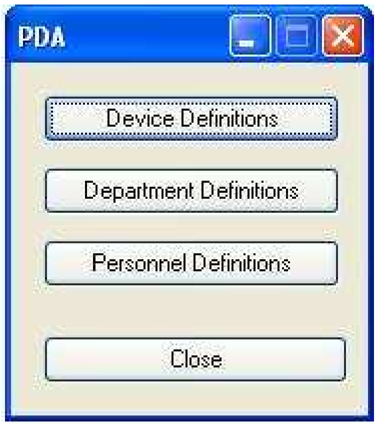
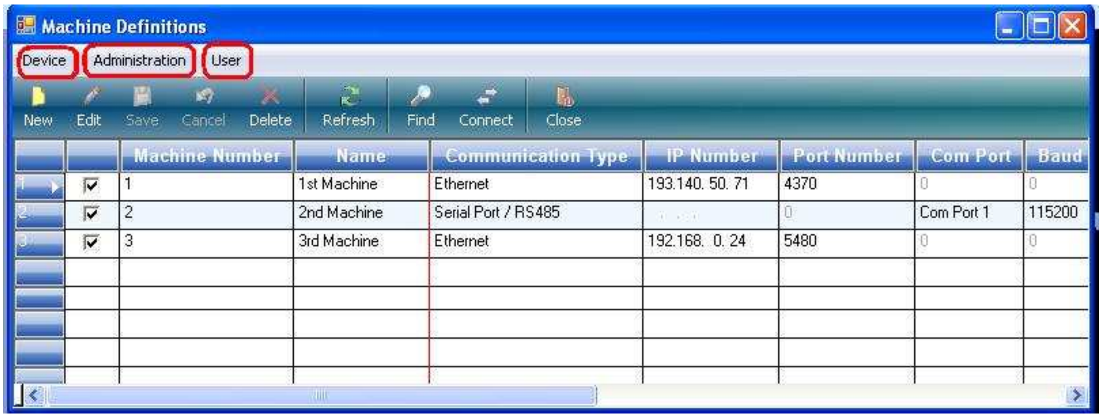


Figure 5.1 Main window of program

5.1 Device Definitions

This feature is used to define communication type (RS232 or Ethernet) and to provide communication between machine and programs. This menu consists of different main icons named device, administration and user. Usage of this feature is very easy. Figure 5.2 shows the menus. Uses of these menus are summarized in the following sentences.



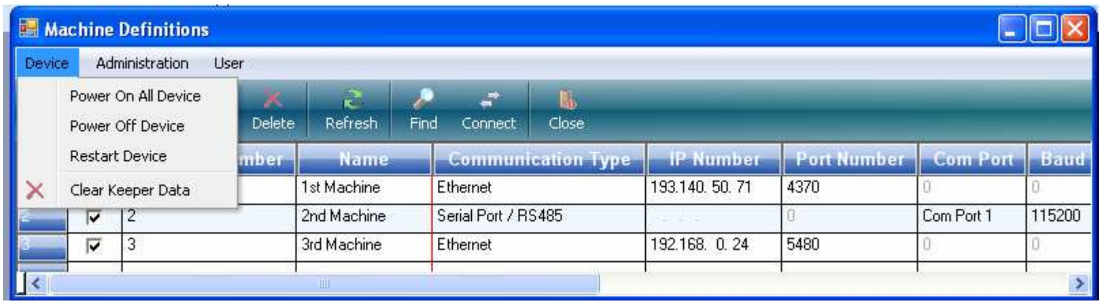
		Machine Number	Name	Communication Type	IP Number	Port Number	Com Port	Baud
1	☑	1	1st Machine	Ethernet	193.140. 50. 71	4370	0	0
2	☑	2	2nd Machine	Serial Port / RS485	. . .	0	Com Port 1	115200
3	☑	3	3rd Machine	Ethernet	192.168. 0. 24	5480	0	0

Figure 5.2 Menus of machine definition

After selection of the communication type, connection is established. If user clicks to red marked menus, the user may execute different transactions such as attendance records, clear logs, power on or off the fingerprint device etc.

If user clicks the **device** menu he/she can see different 4 sub menu. These menus are shown in Figure 5.3.

- Power On All Device : This feature is use to open all devices in the system.
- Power Of Device : This feature is use to shutdown the current device.
- Restart Device : This feature is use to restart the current device.
- Clear Keeper Data : This feature is use to delete saved logs.

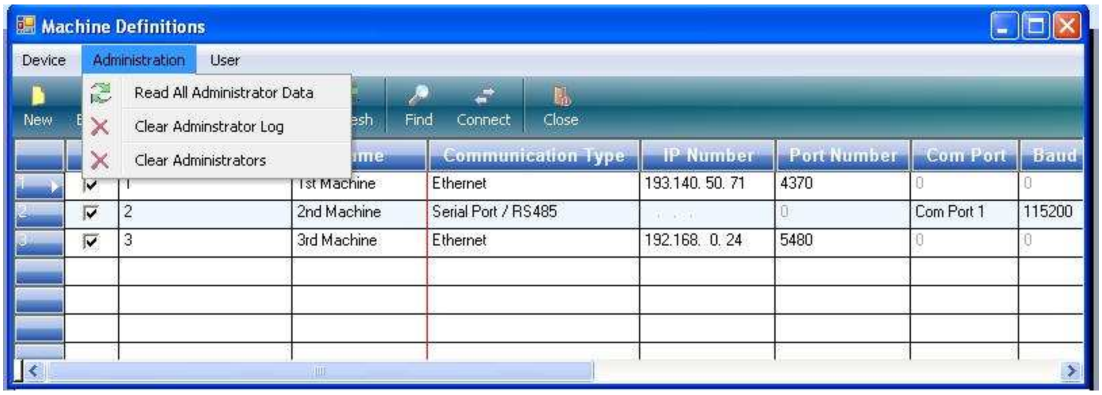


		...mber	Name	Communication Type	IP Number	Port Number	Com Port	Baud
			1st Machine	Ethernet	193.140. 50. 71	4370	0	0
2	☑	2	2nd Machine	Serial Port / RS485	. . .	0	Com Port 1	115200
3	☑	3	3rd Machine	Ethernet	192.168. 0. 24	5480	0	0

Figure 5.3 Menus of device

If user clicks the **administration** menu he/she can see different three sub menu. These menus are shown in Figure 5.4.

- Read All Administrator Data : This feature is use to read administrator data.
- Clear Administrator Log :This feature is use to delete administrator data.
- Clear Administrators : This feature is use to delete administrators.

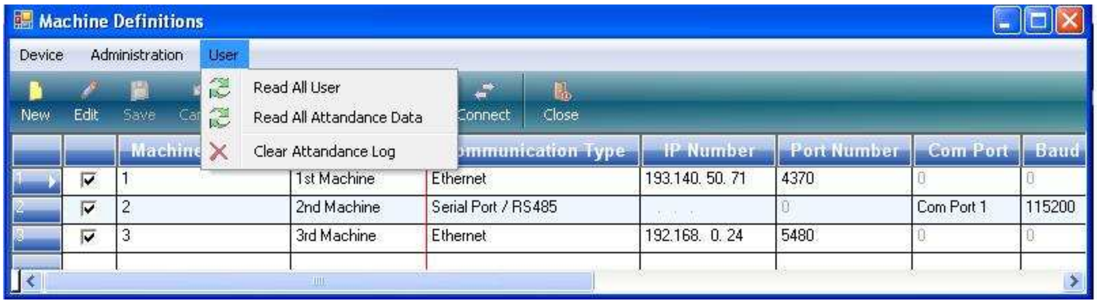


			ime	Communication Type	IP Number	Port Number	Com Port	Baud
1	☑	1	1st Machine	Ethernet	193.140. 50. 71	4370	0	0
2	☑	2	2nd Machine	Serial Port / RS485	. . .	0	Com Port 1	115200
3	☑	3	3rd Machine	Ethernet	192.168. 0. 24	5480	0	0

Figure 5.4 Menus of administration

If user clicks the **user** menu he/she can see different 3 sub menu. These menus are shown in Figure 5.5.

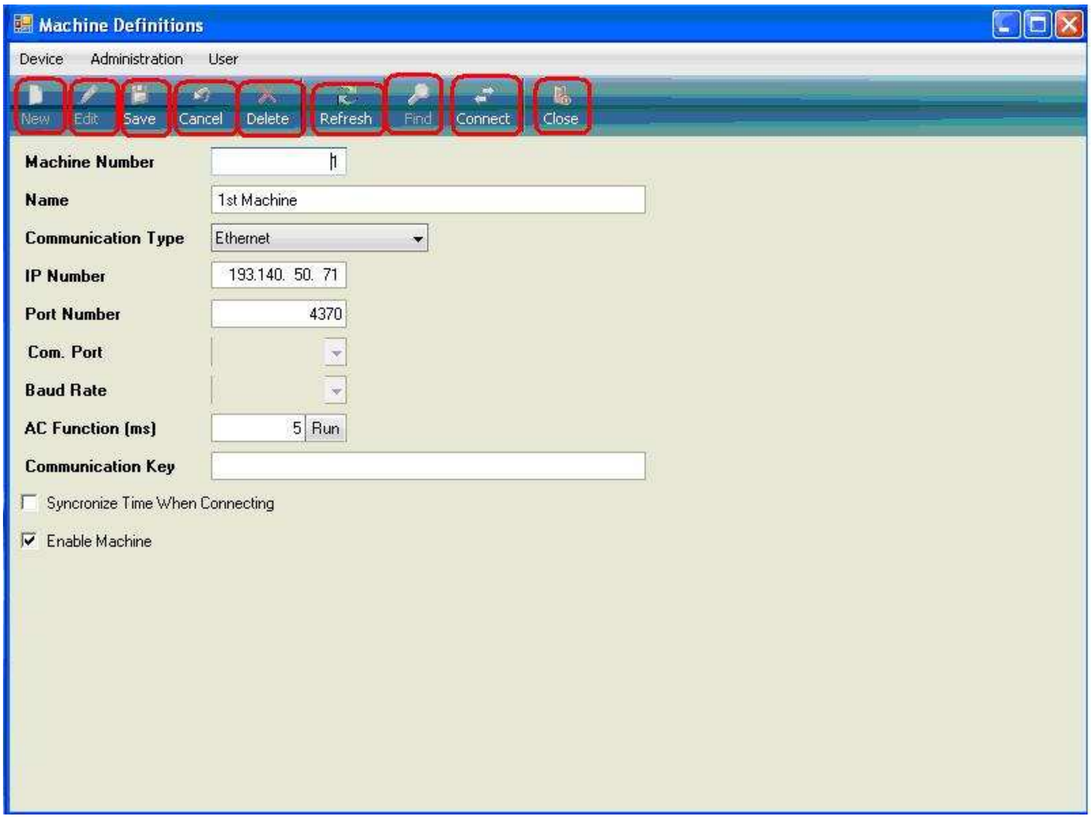
- Read All User : This feature is use to define all user.
- Read All Attendance Data : This feature is use to read all attendance data.
- Clear Attendance Log : This feature is use to delete attendance data.



		Machine		ommunication Type	IP Number	Port Number	Com Port	Baud
1	☑	1	1st Machine	Ethernet	193.140. 50. 71	4370	0	0
2	☑	2	2nd Machine	Serial Port / RS485	. . .	0	Com Port 1	115200
3	☑	3	3rd Machine	Ethernet	192.168. 0. 24	5480	0	0

Figure 5.5 Menus of user

Figure 5.6 shows the machine definition menus. Explanation of these red marked menus and the others are given below.



Machine Number: 1
Name: 1st Machine
Communication Type: Ethernet
IP Number: 193.140. 50. 71
Port Number: 4370
Com. Port:
Baud Rate:
AC Function (ms): 5
Communication Key:
□ Syncronize Time When Connecting
☑ Enable Machine

Figure 5.6 Menus of devices definition

New : To select a new connection type, this icon is use.

Edit : To edit current connection.

Save : To save changed information of current connection.

Cancel : To cancel current connection.

Delete : To delete current connection.

Refresh : To refresh current connection.

Find : To find any connection.

Connect : To connect the fingerprint device.

Close : To close current window.

Machine Number : If the system has more than one machine, machine number must be indicated.

Name : If desired machine is named.

Communication Type: Ethernet or serial must be select.

IP Number **:** If ethernet type connection is select IP number is required. In our system IP number is 193.140.50.71.

Port Number **:** 4370

Com Port **:** Port 1

Boud Rate **:** This menu indicates data flow speed.

Ac Function **:** Relay time.

Communication Key: If password is desired, this box must fill.

Communication types such as serial or TCP/IP communication are summarized as fallowing part.

5.1.1 Serial Communication

Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel. Generally, serial communication is done via RS232 port. Fingerprint device, used in this study, has a serial port. So, if desired, program can provide a communication via serial port with the fingerprint device. However this type of communication has a slower progress than IP communication.

5.1.1.1 RS-232

RS-232 is a standard for serial binary data signals connecting between a DTE (Data Terminal Equipment) and a DCE (Data Circuit-terminating Equipment). It is commonly used in computer serial ports.

Table 5.1 lists commonly-used RS-232 signals and pin assignments and Table 5.2 provides the explanation for some signal used in data communications.

Table 5.1 RS-232 pin assignments

Signal		Origin		DB-25	DE-9 (TIA-574)	EIA/TIA 561	Yost
Name	Abbreviation	DTE	DCE				
Common Ground	G			7	5	4	4,5
Protective Ground	PG			1	-	-	
Transmitted Data	TxD	●		2	3	6	3
Received Data	RxD		●	3	2	5	6
Data Terminal Ready	DTR	●		20	4	3	2
Data Set Ready	DSR		●	6	6	1	7
Request To Send	RTS	●		4	7	8	1
Clear To Send	CTS		●	5	8	7	8
Carrier Detect	DCD		●	8	1	2	7
Ring Indicator	RI		●	22	9	1	-

Table 5.2 The signal meanings in data communication

Signal	Signal Meaning
TD	Serial data output (TXD)
RD	Serial data input (RXD)
RTS	Indicate that the modem is ready for data exchange
CTS	It becomes active when the modem detects "Carrier" signal from the other side of line
DSR	DCE signal that is ready for work
SG	Pin is grounded
CD	DCE informs that line is ready for contact
DTR	Inform DCE device that DTE is ready
RI	Inform when "bell" signals is detected on the phone line

5.1.2 TCP/IP Communications

TCP/IP protocol is used to connect internet and other networks. It consists of two important protocols named the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Generally connection between internet or other networks is done with cat5 or cat6 data cables. Figure 5.7 shows the pin diagrams of RJ45 connector.

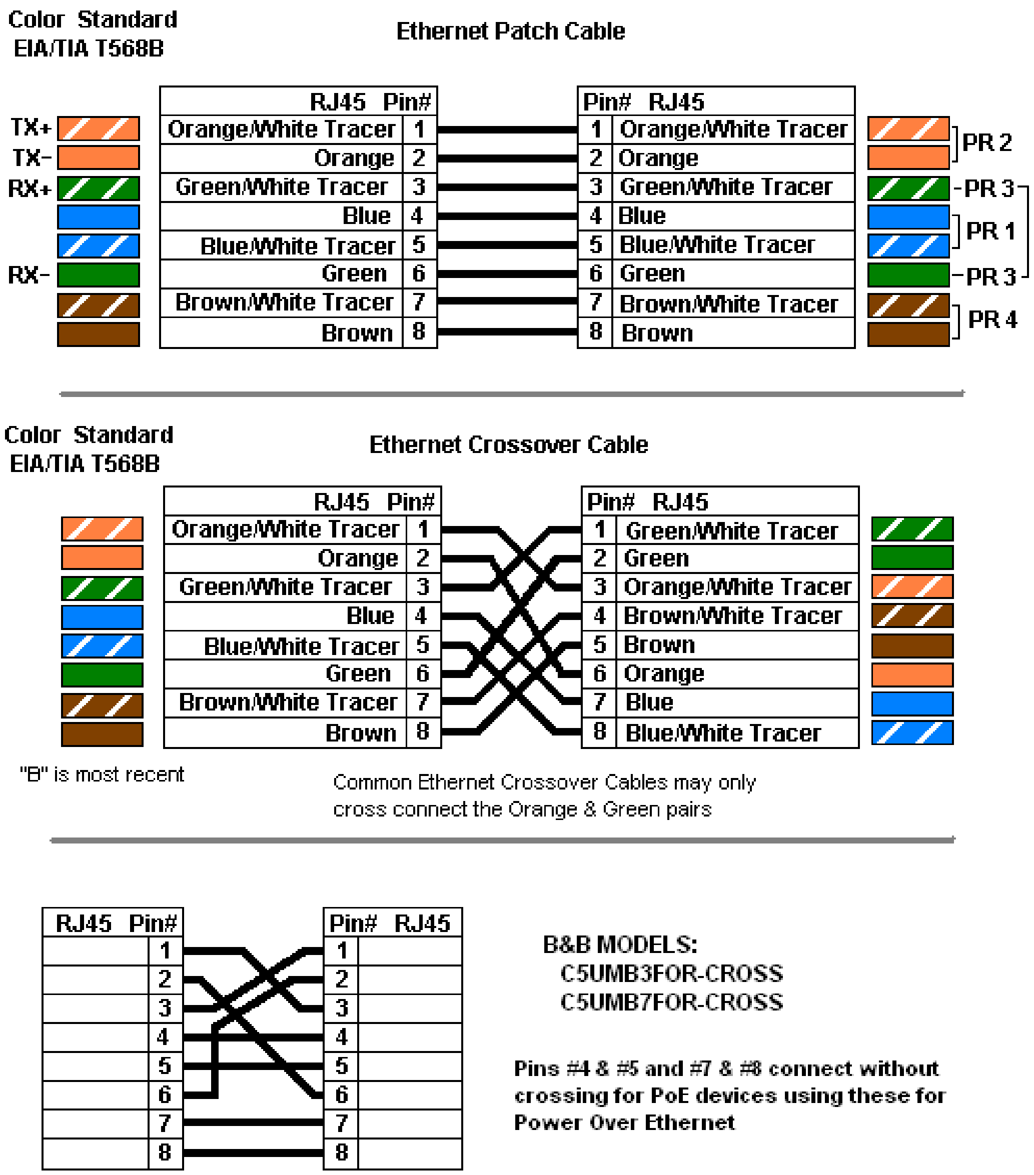
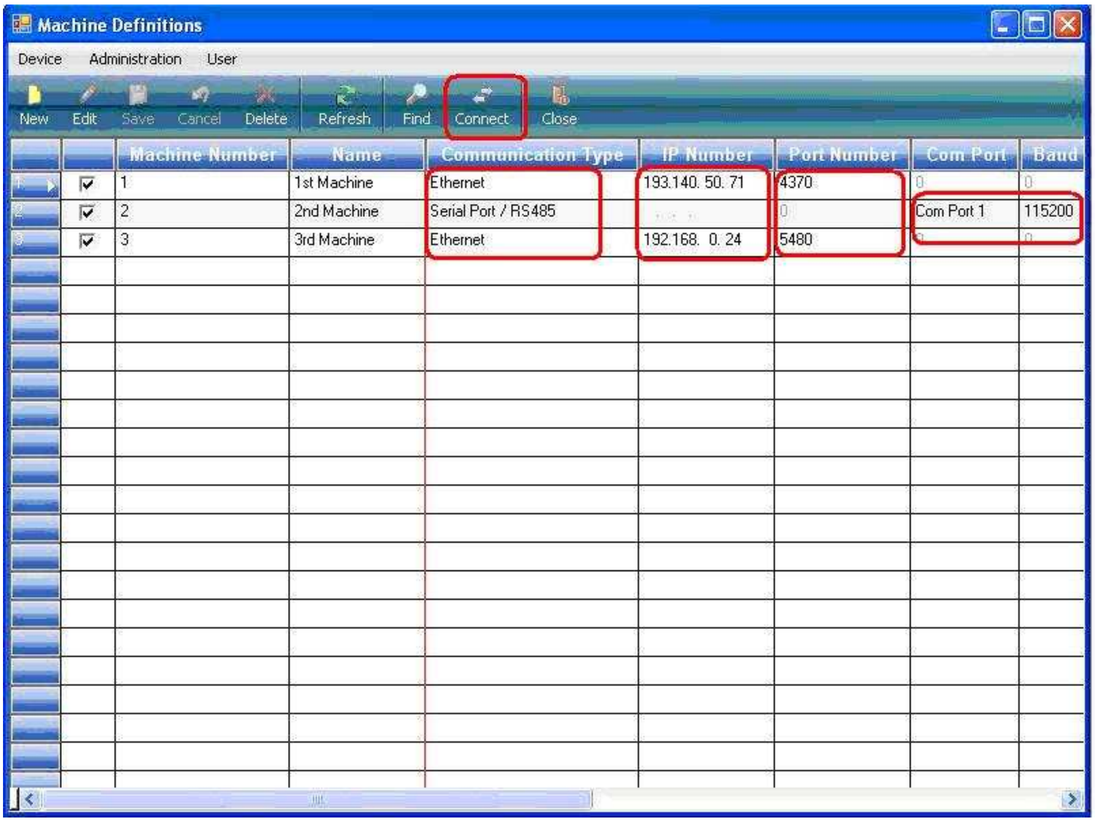


Figure 5.7 Pin diagrams of data cable

After determining the connection type, we can go on with the other interface of machine definition option. Here, filling in the necessary information connection is established. Figure 5.8 shows the communication screen.

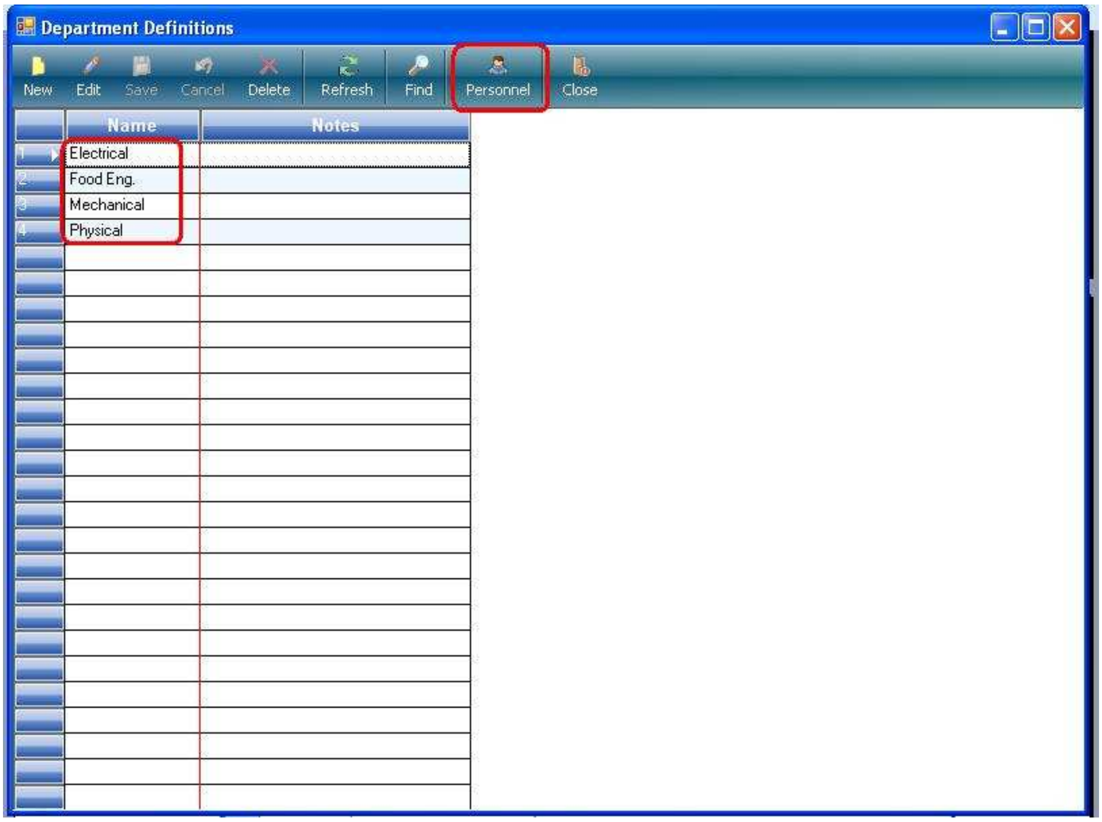


		Machine Number	Name	Communication Type	IP Number	Port Number	Com Port	Baud
1	☑	1	1st Machine	Ethernet	193.140. 50. 71	4370	0	0
2	☑	2	2nd Machine	Serial Port / RS485	. . .	0	Com Port 1	115200
3	☑	3	3rd Machine	Ethernet	192.168. 0. 24	5480	0	0

Figure 5.8 Communication screen

5.2 Department Definitions

This menu includes information about all departments and staff who worked in that department. Using this menu, if desired, you can add or remove the kind of department and staff. If red marked menu (personnel) is clicked which is shown in following figure, then all staff can be listed. In the same manner, in order to arrange department name or notes which belongs to that department, as in the previous menu of device definition, **new**, **edit**, **save**, **cancel**, **delete** and **find** buttons can be used. To close current window **close** button must be use. Figure 5.9 shows the features of this menu.

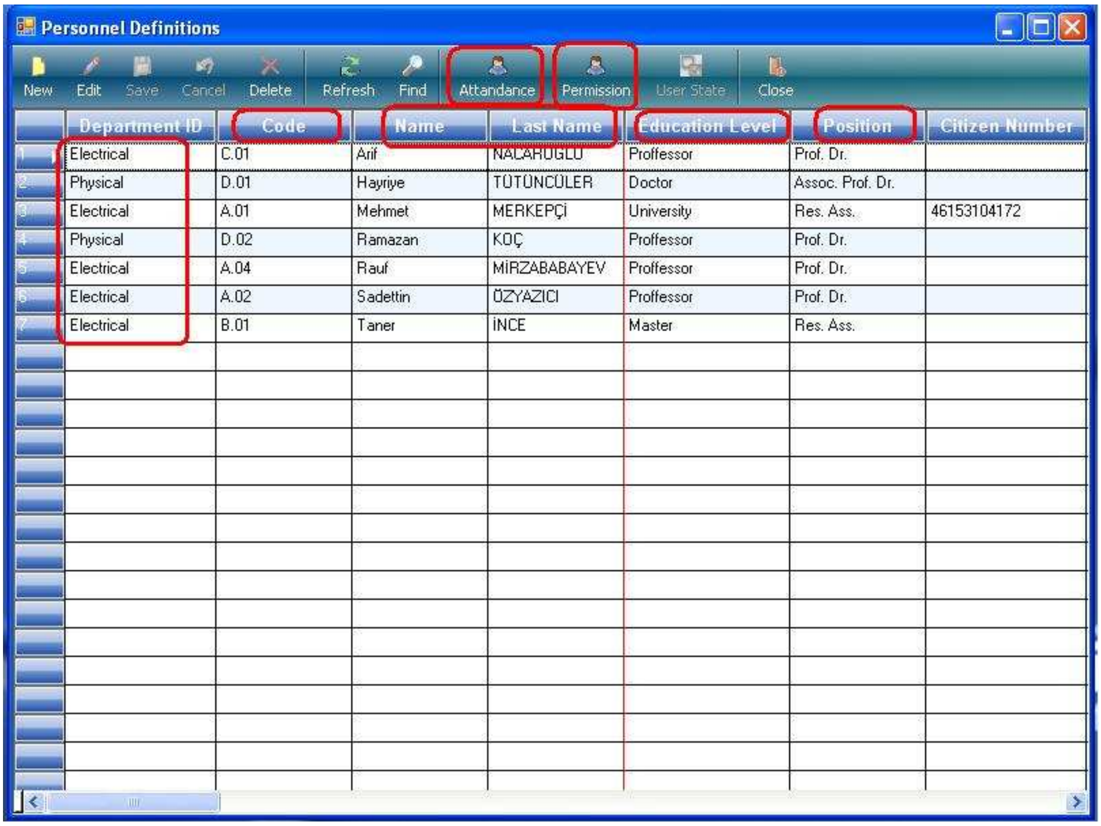


	Name	Notes
1	Electrical	
2	Food Eng.	
3	Mechanical	
4	Physical	

Figure 5.9 Menus of department definition

5.3 Personnel Definition

Using this menu, we can obtain all the activities of any person, such as user state, attendance records (entrance or exit time), permission etc. If desired, attendance records, permission records and user state can be listed by the help of icons which is placed on personnel definition menu. To do this, firstly, you must select number which is belongs to that person and should be click on the related button. This menu is shown in Figure 5.10. Staffs are sorted according to their working department. All staff has a **code** number. Also, all staff may be listed by **department**, **code**, **education**, **position**, **name**, **last name**, **education level**, **citizen number**, **position** etc. and by the help of **attendance** and **permission** buttons, attendance and permission records may be seen belong to any person.

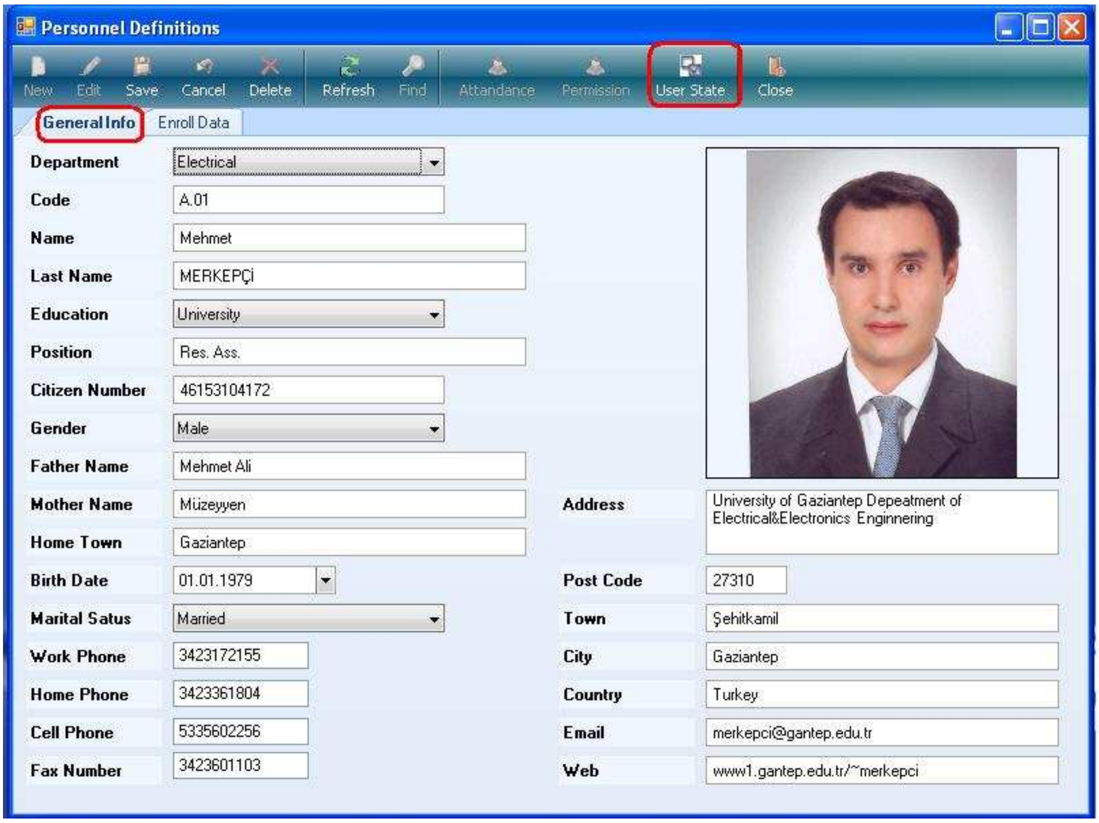


	Department ID	Code	Name	Last Name	Education Level	Position	Citizen Number
1	Electrical	C.01	Arif	NALCAROGLU	Proffessor	Prof. Dr.	
2	Physical	D.01	Hayriye	TÜTÜNCÜLER	Doctor	Assoc. Prof. Dr.	
3	Electrical	A.01	Mehmet	MERKEPÇİ	University	Res. Ass.	46153104172
4	Physical	D.02	Ramazan	KOÇ	Proffessor	Prof. Dr.	
5	Electrical	A.04	Rauf	MİRZABABAYEV	Proffessor	Prof. Dr.	
6	Electrical	A.02	Sadettin	ÖZYAZICI	Proffessor	Prof. Dr.	
7	Electrical	B.01	Taner	İNCE	Master	Res. Ass.	

Figure 5.10 A part of personnel definition menu

In addition, in order to see all information about selected person, his/her number must be double clicked. An example of this situation is shown in Figure 5.11. All information about selected person is listed. If desired, these records are changeable.

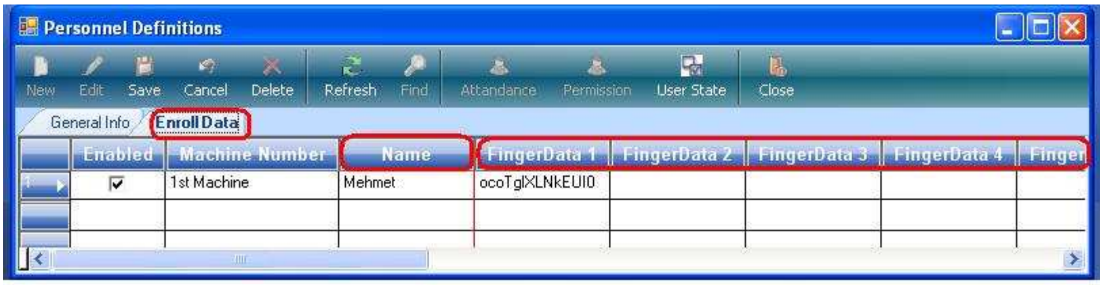
Thanks to **user state** button shown in figure 5.11 user’s situation can be changed. Namely, authorized person’s authorizations may be cancelled. Thus, person’s information may be stored in the fingerprint device but person can not enter the system.



Department: Electrical
Code: A.01
Name: Mehmet
Last Name: MERKEPÇİ
Education: University
Position: Res. Ass.
Citizen Number: 46153104172
Gender: Male
Father Name: Mehmet Ali
Mother Name: Müzeyyen
Home Town: Gaziantep
Birth Date: 01.01.1979
Marital Satus: Married
Work Phone: 3423172155
Home Phone: 3423361804
Cell Phone: 5335602256
Fax Number: 3423601103
Address: University of Gaziantep Depeatment of Electrical&Electronics Enginnering
Post Code: 27310
Town: Şehitkamil
City: Gaziantep
Country: Turkey
Email: merkepci@gantep.edu.tr
Web: www1.gantep.edu.tr/~merkepci

Figure 5.11 A part of personnel definition menu

Here **enroll data** button holds different information about the authorized person. This information includes user state (enable or disable), how recorded the name of person in the fingerprint device, how recorded the finger encoding in the fingerprint device and whether the person authorized for this machine. Because of person may not be authorized for this machine.



	Enabled	Machine Number	Name	FingerData 1	FingerData 2	FingerData 3	FingerData 4	Finger
	☑	1st Machine	Mehmet	ocoTglXLNkEUI0				

Figure 5.12 Enroll data menu

CHAPTER 6

CONCLUSIONS

Nowadays access control systems are nearly using everywhere. Fingerprint based access control can be given as example of this type of system. To prevent undesired situations (theft, absenteeism, lost key etc.) we use fingerprint-based systems in houses, companies, offices, workplaces etc. Especially big companies must have such systems in order to check and observe movements (entrance, exit, attendance, permission etc.) of their employees. Generally such a system is called Access Control and Time Attendance System (AC&TA).

The AC&TA system plays the biggest part to increase productivity. Also, the system is controlled by means of a computer program. Thus, error rate of this system is almost zero. This feature satisfies facility and opportunity for administrators of companies.

The other feature of the designed system is having a door lock. Owing to this feature one can prevent undesired access. By this means, forgery can be prevented.

Consequently, in this study, we overviewed the fingerprint recognition and we developed a program which based on fingerprint recognition. Using this program, each user can be authenticated differently in every different machine. In this way, a user may have access into a door while not having access into others.

REFERENCES

[1] Jain A. K., Hong L., and Pankanti S., (February, 2000). Biometrics: Promising frontiers for emerging identification market, *Commun. ACM*, Vol. 43, No. 2, pp. 91–98.

[2] International Biometric Group, www.biometricgroup.com.

[3] Sheng W, Howells G, Fairhurst M and Deravi F, (September, 2007). A Memetic Fingerprint Matching Algorithm, IEEE Transactions On Information Forensics And Security, Vol. 2, No. 3, pp. 402-412

[4] Jain A. K., Ross A., and Pankanti S., (June, 2006). Biometrics: A tool for information security, *IEEE Trans. Inf. Forensics Security*, Vol. 1, No. 2, pp. 125–143.

[5] Lee H. C. and Gaensslen R. E., (1991). *Advances in Fingerprint Technology*, Elsevier, New York.

[6] Rhodes H. T. F., (1956). *Alphonse Bertillon: Father of Scientific Detection*, Abelard-Schuman, New York.

[7] http://www.biometricscatalog.org/NSTCSubcommittee.

[8] Federal Bureau of Investigation, (1984) *The Science of Fingerprints: Classification and Uses*, U. S.Government Printing Office, Washington, D. C.

[9] www.securco.com/accesscontrolsystems.htm.

[10] http://automated-machines.com/automated-time-attendance-system.html.

[11] Kawagoe M. and Tojo A., (1984). Fingerprint Pattern Classification, *Pattern Recognition*, Vol. 17, No. 3, pp. 295-303.

[12] http://en.wikipedia.org/wiki/Card_reader.

[13] Bovik, Al, (2000). *Handbook of Image and Video processing*, USA, Academic Press.

[14] Young N. D., Harkin G., Bunn R. M., McCulloch D. J. and R. Wilks W. and A. Knapp G., (January, 1997). Novel Fingerprint Scanning Arrays Using Polysilicon TFT's on Glass and Polymer Substrates, *IEEE Electron Device Letters*, Vol. 18, No. 1, pp. 19–20.

[15] Mehtre B. M. and Chatterjee B., (1989). Segmentation of fingerprint *Pattern Recognition*, Vol. 22, No. 4, pp. 381–385.

[16] Ratha N., Karu K., Chen S. and Jain A. K., (1996). A Real-time Matching System for Large Fingerprint Database, *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. 18, No. 8, pp. 799-813.

[17] Jain A., Hong L., Pankanti S., and Bolle R., (September 1997). On-line identity-authentication system using fingerprints, *Proceedings of IEEE (Special Issue on Automated Biometrics)*, vol. 85, pp. 1365– 1388.

[18] http://www.perkotek.com.tr/parmakizi.htm?gclid=CNWJnOT01poCFYYVzAo dTE0Usw

[19] http://hk.zksoftware.com/english/product_e_zem100edk.asp.

[20] Stutz Michael, (2000). Conventional versus electron flow, *All About Circuits*, Vol. 1, Chapter 1.

APPENDIX A

SOME PROGRAMMING CODES FROM THE MAIN PROGRAM

Get Data from Database Sample (Attendance Form)

```
private void RefreshData() {
  string sql;

  //Filter For A Personnel Or Get Whole Attandance Data Between Selected Date Range
  if (this.PersonnelID > 0)
    sql = "SELECT g.[GLogID], g.[MachineNumber], g.[EnrollNumber], g.[InOutMode],
g.[DateTime] FROM [tblGLog] g LEFT JOIN [tblEnrollData] e ON e.[EnrollNumber] =
g.[EnrollNumber] WHERE e.[PersonnelID] = " + PersonnelID.ToString() + " AND
g.[DateTime] >= #" + dtpStartDate.Value.ToString().Replace(".", "/") + "# AND
g.[DateTime] <= #" + dtpEndDate.Value.ToString().Replace(".", "/") + "# ORDER BY
g.[DateTime] DESC";
  else
    sql = "SELECT [GLogID], [MachineNumber], [EnrollNumber], [InOutMode], [DateTime]
FROM [tblGLog] WHERE [DateTime] >= #" + dtpStartDate.Value.ToString().Replace(".",
"/") + "# AND [DateTime] <= #" + dtpEndDate.Value.ToString().Replace(".", "/") + "#
ORDER BY [DateTime] DESC";

    this.grdAttandance.DataSource = null;
    this.grdAttandance.ResetLayouts();

    DataTable dTable = MainForm.FillTable(sql, "tblGLog");

    this.grdAttandance.Rows.ColumnFilters.ClearAllFilters();
    this.grdAttandance.DataSource = dTable;

    this.ShowTools(false);
}
```

Filling Combo Box Sample (Personnel Form)

```
private void FillDepartment(string ValueListName) {
  try {
    this.cmbDepartment.BeginUpdate();
    this.cmbDepartment.Items.Clear();

    DataTable dTable = MainForm.FillTable("SELECT [DepartmentID], [Name] FROM
[tblDepartment] ORDER BY [Name]", "tblDepartment");

    if (!this.grdPersonnel.DisplayLayout.ValueLists.Exists(ValueListName))
      this.grdPersonnel.DisplayLayout.ValueLists.Add(ValueListName);

    this.grdPersonnel.DisplayLayout.ValueLists[ValueListName].ValueListItems.Clear();

    if (dTable != null) {
      foreach (DataRow dRow in dTable.Rows) {
        this.cmbDepartment.Items.Add(new ValueListItem(dRow["DepartmentID"],
dRow["Name"].ToString()));


this.grdPersonnel.DisplayLayout.ValueLists[ValueListName].ValueListItems.Add(new
ValueListItem((int)dRow["DepartmentID"], dRow["Name"].ToString()));
       }
    }
  }
  finally {
    this.cmbDepartment.EndUpdate();
  }
}
```

Adding Relation Between Enroll Data and Personnel Data (Enrollment Form)

```
private void cmdAccept_Click(object sender, EventArgs e) {
  foreach (UltraGridRow Row in this.grdEnrollData.Rows) {
    if (Row.Selected) {
      Int16 EnrollNumber = (Int16)Row.Cells["EnrollNumber"].Value;

      try {
        string sql = "UPDATE [tblEnrollData] SET [PersonnelID] = {1} WHERE
[EnrollNumber] = {0}";

        sql = string.Format(sql, EnrollNumber, this.PersonnelID);

        if (MainForm.ExecNonQuery(sql) > 0)
          MessageBox.Show("Personnel Relation Established Succesfully", "Personnel
Definition", MessageBoxButtons.OK, MessageBoxIcon.Information);
        else
          MessageBox.Show("Unable To Establish Personnel Relation!", "Personnel
Definition", MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
      catch (Exception ex) {
        MessageBox.Show("Unexpected Error Occured!\n\n" + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
      }

      this.DialogResult = DialogResult.OK;
    }
  }
}
```

Load Personnel Image (Image Form)

```
private void cmdLoadImage_Click(object sender, EventArgs e) {
  OpenFileDialog oDialog = new OpenFileDialog();

  oDialog.AddExtension = false;
  oDialog.CheckFileExists = true;
  oDialog.CheckPathExists = true;
  oDialog.DefaultExt = "jpg";
  oDialog.Filter = "Compu Serve Graphics Interchange (*.gif)|*.gif|" +
                     "JPEG-JIF Compliant (*.jpg, *.jif, *.jpeg)|*.jpg;*.jif;*.jpeg|"
+
                     "Portable Network Graphics (*.png)|*.png|" +
                     "Windows Enhanced Meta File (*.emf)|*.emf|" +
                     "Windows Meta File (*.wmf)|*.wmf|" +
                     "Windows or OS/2 Bitmap (*.bmp)|*.bmp";
  oDialog.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop);
  oDialog.Multiselect = false;
  oDialog.RestoreDirectory = true;
  oDialog.Title = "Load Image...";

  if (oDialog.ShowDialog(this) == DialogResult.OK) {
    Image img = null;

    try {
      img = System.Drawing.Image.FromFile(oDialog.FileName);

      this.Image = new Bitmap(img);
    }
    catch (Exception ex) {
      MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
    finally {
      if (img == null)
        img.Dispose();
    }
  }
}
```

Connect To Machine (Machine Form)

```
private bool ConnectToDevice() {
 UltraGridRow Row = this.grdMachine.ActiveRow;

   if (Row != null && Row.IsDataRow) {
      try {
        bool dataChanged = Row.DataChanged;

        this.grdMachine.EventManager.AllEventsEnabled = false;

        if (this.CheckMachine()){
          int acFun = 0;

          if
((MainForm.CommunicationTypes)((byte)Row.Cells["CommunicationTypeID"].Value) ==
MainForm.CommunicationTypes.Ethernet) {
            if
(!MainForm.ConnectToNet(Row.Cells["IPNumber"].Value.ToString().Replace(" ", ""),
(System.Int16)Row.Cells["Port"].Value))
              MessageBox.Show("Unable To Connect Machine Via Ethernet!", "Machine
Definition", MessageBoxButtons.OK, MessageBoxIcon.Error);
            else {
              MessageBox.Show("Connected To Machine Succesfully!", "Machine
Definition", MessageBoxButtons.OK, MessageBoxIcon.Information);
              Row.Cells["Connected"].Value = 1;

              if (MainForm.GetACFun(ref acFun))
                Row.Cells["ACFun"].Value = acFun;

              if (!dataChanged)
                Row.Update();

                return true;
              }
            }
            else {
              if (!MainForm.ConnectToComPort((byte)Row.Cells["ComPort"].Value,
(System.Int16)Row.Cells["MachineNumber"].Value, (int)Row.Cells["BaudRate"].Value))
                MessageBox.Show("Unable To Connect Machine Via Com Port!", "Machine
Definition", MessageBoxButtons.OK, MessageBoxIcon.Error);
              else {
                MessageBox.Show("Connected To Machine Succesfully!", "Machine
Definition", MessageBoxButtons.OK, MessageBoxIcon.Information);
                Row.Cells["Connected"].Value = 1;

            if (MainForm.GetACFun(ref acFun))
              Row.Cells["ACFun"].Value = acFun;

            if (!dataChanged)
              Row.Update();

            return true;
          }
        }
      }

      Row.Cells["Connected"].Value = 0;

      if (!dataChanged)
        Row.Update();
    }
    finally {
      this.grdMachine.EventManager.AllEventsEnabled = true;
    }
  }

  return false;
}
```

Power off Machine (Machine Form)

```
private void PowerOffDevice(){
  UltraGridRow Row = this.grdMachine.ActiveRow;
  if (Row != null && Row.IsDataRow) {
    if ((int)Row.Cells["Connected"].Value == 1) {
      if (MessageBox.Show("The Device Will Be Powered Off\n\nDo You Want To
Continue?", "Machine Definition", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
== DialogResult.Yes)
        if (!MainForm.PowerOffDevice((System.Int16)Row.Cells["MachineNumber"].Value))
          MessageBox.Show("Unable To Powered Off The Device!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
      MessageBox.Show("Device Is Not Connected!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
  }
}
```

Restart Machine (Machine Form)

```
private void RestartDevice(){
  UltraGridRow Row = this.grdMachine.ActiveRow;

  if (Row != null && Row.IsDataRow) {
    if ((int)Row.Cells["Connected"].Value == 1) {
      if (MessageBox.Show("The Device Will Be Restarted\n\nDo You Want To Continue?",
"Machine Definition", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
        if (!MainForm.RestartDevice((System.Int16)Row.Cells["MachineNumber"].Value))
{
          MessageBox.Show("Unable To Restart The Device!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

          try {
            this.grdMachine.EventManager.AllEventsEnabled = false;

            Row.Cells["Connected"].Value = 0;
            Row.Update();
          }
          finally {
            this.grdMachine.EventManager.AllEventsEnabled = true;
          }
        }
    }
    else
      MessageBox.Show("Device Is Not Connected!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
  }
}
```

Get All User from Machine (Machine Form)

```
private void ReadAllUserID() {
  UltraGridRow Row = this.grdMachine.ActiveRow;

  if (Row != null && Row.IsDataRow) {
    if ((int)Row.Cells["Connected"].Value == 1) {
      int machineNumber = (System.Int16)Row.Cells["MachineNumber"].Value;
      MachineUser[] Users = null;

      if (MainForm.ReadAllUserID(machineNumber)) {
        int counter = 0;
        int enrollNumber = 0;
        string name = "";
        string password = "";
        int privilige = 0;
        bool enable = false;

        while (MainForm.GetAllUserInfo(machineNumber, ref enrollNumber, ref name, ref
password, ref privilige, ref enable)) {
          counter++;
          Array.Resize(ref Users, counter);
          Array.Resize(ref Users[counter - 1].FingerIndex, 10);

          Users[counter - 1].Enabled = enable;
          Users[counter - 1].EnrollNumber = enrollNumber;
          Users[counter - 1].MachineNumber = machineNumber;
          Users[counter - 1].Name = name;
          Users[counter - 1].Password = password;
          Users[counter - 1].Privilige = privilige;
          string tmpData = "";
          int tmpLength = 0;

          for (int fingerIndex = 0; fingerIndex < 10; fingerIndex++) {
            if (MainForm.GetUserTmpStr(machineNumber, enrollNumber, fingerIndex, ref
tmpData, ref tmpLength))
              Users[counter - 1].FingerIndex[fingerIndex] = tmpData;
          }
        }

        string insertEnrollSQL = "INSERT INTO [tblEnrollData] ([MachineNumber],
[EnrollNumber], [Name], [FingerData1], [FingerData2], [FingerData3], [FingerData4],
[FingerData5], [FingerData6], [FingerData7], [FingerData8], [FingerData9],
[FingerData10], [Password], [Privilige], [Enabled]) VALUES ({0}, {1}, '{2}', '{3}',
'{4}', '{5}', '{6}', '{7}', '{8}', '{9}', '{10}', '{11}', '{12}', '{13}', {14},
{15})";
        string updateEnrollSQL = "UPDATE [tblEnrollData] SET [Name] = '{2}',
[FingerData1] = '{3}', [FingerData2] = '{4}', [FingerData3] = '{5}', [FingerData4] =
'{6}', [FingerData5] = '{7}', [FingerData6] = '{8}', [FingerData7] = '{9}',
[FingerData8] = '{10}', [FingerData9] = '{11}', [FingerData10] = '{12}', [Password] =
'{13}', [Privilige] = {14}, [Enabled] = {15} WHERE [MachineNumber] = {0} AND
[EnrollNumber] = {1}";
        string sqlEnroll;

        if (Users != null) {
          object oldEnrollNumber = 0;
          object oldFingerIndex = 0;

          for (int index = 0; index < Users.Length; index++) {
            oldEnrollNumber = MainForm.GetDBValue("SELECT [EnrollNumber] FROM
[tblEnrollData] WHERE [MachineNumber] = " + Users[index].MachineNumber + " AND
[EnrollNumber] = " + Users[index].EnrollNumber);

            if (oldEnrollNumber == null || (System.Int16)oldEnrollNumber == 0)
              sqlEnroll = string.Format(insertEnrollSQL,
(System.Int16)Users[index].MachineNumber, (System.Int16)Users[index].EnrollNumber,
Users[index].Name, Users[index].FingerIndex[0], Users[index].FingerIndex[1],
Users[index].FingerIndex[2], Users[index].FingerIndex[3],
Users[index].FingerIndex[4], Users[index].FingerIndex[5],
Users[index].FingerIndex[6], Users[index].FingerIndex[7],
Users[index].FingerIndex[8], Users[index].FingerIndex[9], Users[index].Password,
(byte)Users[index].Privilige, Users[index].Enabled);
            else
              sqlEnroll = string.Format(updateEnrollSQL,
(System.Int16)Users[index].MachineNumber, (System.Int16)Users[index].EnrollNumber,
Users[index].Name, Users[index].FingerIndex[0], Users[index].FingerIndex[1],
Users[index].FingerIndex[2], Users[index].FingerIndex[3],
Users[index].FingerIndex[4], Users[index].FingerIndex[5],
Users[index].FingerIndex[6], Users[index].FingerIndex[7],
Users[index].FingerIndex[8], Users[index].FingerIndex[9], Users[index].Password,
(byte)Users[index].Privilige, Users[index].Enabled);

            MainForm.ExecNonQuery(sqlEnroll);
          }
        }
      }
    }
    else
      MessageBox.Show("Device Is Not Connected!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
  }
}
```

Get All User Attendance Data from Machine (Machine Form)

```
private void ReadAllGLog() {
  UltraGridRow Row = this.grdMachine.ActiveRow;

  if (Row != null && Row.IsDataRow) {
    if ((int)Row.Cells["Connected"].Value == 1) {
      if (MainForm.ReadAllGLogData((System.Int16)Row.Cells["MachineNumber"].Value)) {
        string sql = "INSERT INTO [tblGLog] ([MachineNumber], [EnrollNumber],
[VerifyMode], [InOutMode], [DateTime]) VALUES ({0}, {1}, {2}, {3}, '{4}')";
```

```
        string tmp;
        DateTime date;
        int machineNumber = 0;
        int enrollNumber = 0;
        int verifyMode = 0;
        int inOutMode = 0;
        int year = 0;
        int month = 0;
        int day = 0;
        int hour = 0;
        int minute = 0;

        while
(MainForm.GetAllGLogData((System.Int16)Row.Cells["MachineNumber"].Value, ref
machineNumber, ref enrollNumber, ref machineNumber, ref verifyMode, ref inOutMode,
ref year, ref month, ref day, ref hour, ref minute)) {
          date = new DateTime(year, month, day, hour, minute, 0);
          tmp = string.Format(sql, (System.Int16)machineNumber,
(System.Int16)enrollNumber, (byte)verifyMode, (byte)inOutMode, date);

          MainForm.ExecNonQuery(tmp);
        }

        MessageBox.Show("Attandance Log Saved Succesfully", "Machine Definition",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        MainForm.ClearGLog((System.Int16)Row.Cells["MachineNumber"].Value);
      }
    }
    else
      MessageBox.Show("Device Is Not Connected!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

    AttandanceForm fAttandance = new AttandanceForm();

    fAttandance.ShowDialog(this);
    fAttandance.Close();
    fAttandance = null;
  }
}
```

Delete Data Sample (Machine Form)

```
private void grdMachine_BeforeRowsDeleted(object sender, BeforeRowsDeletedEventArgs
e) {
  e.DisplayPromptMsg = false;

  if (MessageBox.Show("Do You Want To Delete Selected Machine Data!", "Machine
Definition", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
    e.Cancel = true;
  else {
    try {
      string sql = "DELETE FROM [tblMachine] WHERE [MachineNumber] = {0}";

      sql = string.Format(sql,
(System.Int16)e.Rows[0].Cells["MachineNumber"].OriginalValue);

      if (MainForm.ExecNonQuery(sql) > 0)
        MessageBox.Show("Data Deleted Succesfully", "Machine Definition",
MessageBoxButtons.OK, MessageBoxIcon.Information);
      else {
        MessageBox.Show("Unable To Delete Data!", "Machine Definition",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        e.Cancel = true;
      }
    }
    catch {
      MessageBox.Show("Unexpected Error Occured!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      e.Cancel = true;
    }
  }
}
```

Enable / Disable User (Machine Form)

```
private void UserState() {
  UltraGridRow Row = this.grdEnrollData.ActiveRow;

  if (Row != null && Row.IsDataRow) {
    try {
      this.grdEnrollData.EventManager.AllEventsEnabled = false;
      Row.Cells["Enabled"].Value = !((bool)Row.Cells["Enabled"].Value);

      string sql = "UPDATE [tblEnrollData] SET [Enabled] = {1} WHERE [EnrollNumber] =
{0}";
      Int16 machineNumber = (Int16)Row.Cells["MachineNumber"].Value;
      Int16 enrollNumber = (Int16)Row.Cells["EnrollNumber"].Value;

      sql = string.Format(sql, enrollNumber, (bool)Row.Cells["Enabled"].Value);

      if (MainForm.ExecNonQuery(sql) > 0) {
        Row.Update();

        if (machineNumber > 0 && this.ConnectToDevice(machineNumber)) {
          if (MainForm.EnableUser(machineNumber, enrollNumber, machineNumber,
MainForm.FingerIndex.Complete, (bool)Row.Cells["Enabled"].Value))
            MessageBox.Show("User Data Saved Succesfully", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Information);
          else
            MessageBox.Show("Unable To Sent User Data To Device!", "Personnel
Definition", MessageBoxButtons.OK, MessageBoxIcon.Error);

            MainForm.RefreshData(machineNumber);
            MainForm.Disconnect(machineNumber, 30);
        }
        else
          MessageBox.Show("Unable To Connect To Device!", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
      else
        MessageBox.Show("Unable To Save User Data!", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch (Exception ex) {
      MessageBox.Show("Unexpected Error Occured!\n\n" + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    finally {
      this.grdEnrollData.EventManager.AllEventsEnabled = true;
    }
  }
}
```

Saving User Data to Database and Machine (Personnel Form)

```
private void grdEnrollData_BeforeRowUpdate(object sender, CancelableRowEventArgs e) {
  if (!e.Row.DataChanged)
    return;

  e.Cancel = !this.CheckPersonnel();

  if (!e.Cancel) {
    try {
      string sql = "UPDATE [tblEnrollData] SET [Name] = '{1}', [FingerData1] = '{2}',
[FingerData2] = '{3}', [FingerData3] = '{4}', [FingerData4] = '{5}', [FingerData5] =
'{6}', [FingerData6] = '{7}', [FingerData7] = '{8}', [FingerData8] = '{9}',
[FingerData9] = '{10}', [FingerData10] = '{11}', [Password] = '{12}', [Privilige] =
{13}, [Enabled] = {14} WHERE [EnrollNumber] = {0}";
      Int16 machineNumber = (Int16)e.Row.Cells["MachineNumber"].Value;
      Int16 enrollNumber = (Int16)e.Row.Cells["EnrollNumber"].Value;

      sql = string.Format(sql, enrollNumber, e.Row.Cells["Name"].Value.ToString(),
e.Row.Cells["FingerData1"].Value.ToString(),
e.Row.Cells["FingerData2"].Value.ToString(),
e.Row.Cells["FingerData3"].Value.ToString(), e.Row.Cells["FingerData4"].Value,
e.Row.Cells["FingerData5"].Value.ToString(),
e.Row.Cells["FingerData6"].Value.ToString(), e.Row.Cells["FingerData7"].Value,
e.Row.Cells["FingerData8"].Value.ToString(),
e.Row.Cells["FingerData9"].Value.ToString(),
e.Row.Cells["FingerData10"].Value.ToString(),
e.Row.Cells["Password"].Value.ToString(), (byte)e.Row.Cells["Privilige"].Value,
(bool)e.Row.Cells["Enabled"].Value);

      if (MainForm.ExecNonQuery(sql) > 0) {
        if (machineNumber > 0 && this.ConnectToDevice(machineNumber)) {
          bool result = (MainForm.SetUserInfo(machineNumber, enrollNumber,
e.Row.Cells["Name"].Value.ToString(), e.Row.Cells["Password"].Value.ToString(),
(byte)e.Row.Cells["Privilige"].Value, (bool)e.Row.Cells["Enabled"].Value));

          for (int index = 0; index < 10 && result == true; index++)
            result = MainForm.SetUserTmpStr(machineNumber, enrollNumber, index,
e.Row.Cells["Finger" + (index + 1).ToString()].Value.ToString());

          if (result)
            MessageBox.Show("User Data Saved Succesfully", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Information);
          else
            MessageBox.Show("Unable To Sent User Data To Device!", "Personnel
Definition", MessageBoxButtons.OK, MessageBoxIcon.Error);

          MainForm.RefreshData(machineNumber);
          MainForm.Disconnect(machineNumber, 30);
        }
        else
          MessageBox.Show("Unable To Connect To Device!", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
      else {
                MessageBox.Show("Unable To Save User Data!", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        e.Cancel = true;
      }
    }
    catch (Exception ex) {
      MessageBox.Show("Unexpected Error Occured!\n\n" + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
      e.Cancel = true;
    }
  }
}
```

Delete User Data from Database and Machine (Personnel Form)

```
private void grdEnrollData_BeforeRowsDeleted(object sender,
BeforeRowsDeletedEventArgs e) {
  e.DisplayPromptMsg = false;

  if (MessageBox.Show("Do You Want To Delete Selected User Data!", "Personnel
Definition", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
    e.Cancel = true;
  else {
    try {
      string sql = "DELETE FROM [tblEnrollData] WHERE [EnrollNumber] = {0}";
      Int16 machineNumber = (Int16)e.Rows[0].Cells["MachineNumber"].Value;
      Int16 enrollNumber = (Int16)e.Rows[0].Cells["EnrollNumber"].Value;

      sql = string.Format(sql, enrollNumber);

      if (MainForm.ExecNonQuery(sql) > 0) {
        if (this.ConnectToDevice(machineNumber)) {
          if (MainForm.DelUserTmp(machineNumber, enrollNumber,
MainForm.FingerIndex.Complete))
            MessageBox.Show("User Data Deleted Succesfully", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Information);
          else
            MessageBox.Show("Unable To Delete User Data From Device!", "Personnel
Definition", MessageBoxButtons.OK, MessageBoxIcon.Error);

          MainForm.RefreshData(machineNumber);
          MainForm.Disconnect(machineNumber, 30);
        }
      }
      else {
        MessageBox.Show("Unable To Delete User Data!", "Personnel Definition",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        e.Cancel = true;
      }
    }
    catch {
      MessageBox.Show("Unexpected Error Occured!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      e.Cancel = true;
    }
  }
}
```

APPENDIX B

PROGRAM STRUCTURE TABLES

In the appendix, some tables are given based on access records.

- ***tblDepartment:*** This table holds department data.

Table B.1 Table of tblDepartement

Department ID	Automatic Number
Name	Varchar (50)
Notes	Varchar (255)

- ***tblEnrollData:*** This table holds entrance and exit data.

Table B.2 Table of EnrollData

Enroll Number	Int16
Machine Number	Int16
Personnel ID	Int32 (integer)
Name	Varchar (150)
Finger Data 1	Text
Finger Data 2	Text
Finger Data 3	Text
Finger Data 4	Text
Finger Data 5	Text
Finger Data 6	Text
Finger Data 7	Text
Finger Data 8	Text
Finger Data 9	Text
Finger Data 10	Text
Password	Varchar (50)
Privilege	Byte
Enabled	Bit

- ***tblGlog :*** This table arranges to Glog data.

Table B.3 Table of tblGlog

Glog ID	Automatic Number
Machine Number	Int16
Enroll Number	Int16
Verify Mode	Byte
In / Out Mode	Byte
Date / Time	Date / Time

- ***tblMachine :*** This table holds machine's information.

Table B.4 Table of tblMachine

Machine Number	Number
Name	Varchar (50)
Communication Type ID	Byte
IP Number	Varchar (20)
Port	Int16
Com Port	Byte
Baud Rate	Int32 (integer)
Synchronize Time	Bit
Communication Key	Varchar (50)
AC Fun	Int16
Enabled	Bit

- ***tblPermission :*** This table holds permission data.

Table B.5 Table of tblPermission

Permission ID	Automatic Number
Personnel ID	Int32 (integer)
Permission Type ID	Byte
Start Date	Date / Time
End Date	Date / Time

- ***tblPersonnel :*** This table holds personnel data.

Table B.6 Table of tblPersonnel

Personnel ID	Auto Integer
Department ID	Int32 (integer)
Code	Varchar (20)
Name	Varchar (50)
Last Name	Varchar (50)
Education Level	Byte
Position	Varchar (50)
Citizen Number	Varchar (20)
Gender	Byte
Father Name	Varchar (50)
Mother Name	Varchar (50)
Home Town	Varchar (50)
Birth Day	Date / Time
Marital Status	Byte
Work Phone	Varchar (20)
Home Phone	Varchar (20)
Cell Phone	Varchar (20)
Fax Number	Varchar (20)
Address	Varchar (255)
Post Code	Varchar (5)
Town	Varchar (50)
City	Varchar (50)
Country	Varchar (50)
E-mail	Varchar (100)
Web	Varchar (100)
Image	Binary

- ***tblSlog :*** This table holds Slog data.

Table B.7 Table of tblSlog

Slog ID	Automatic Number
Machine Number	Int16
Enroll Number	Int16
Params 1	Int32 (integer)
Params 2	Int32 (integer)
Params 3	Int32 (integer)
Params 4	Int32 (integer)
Manipulation	Byte
Date / Time	Date / Time

The relation between above table can be shown following in Figure B.1 and Figure B.2.

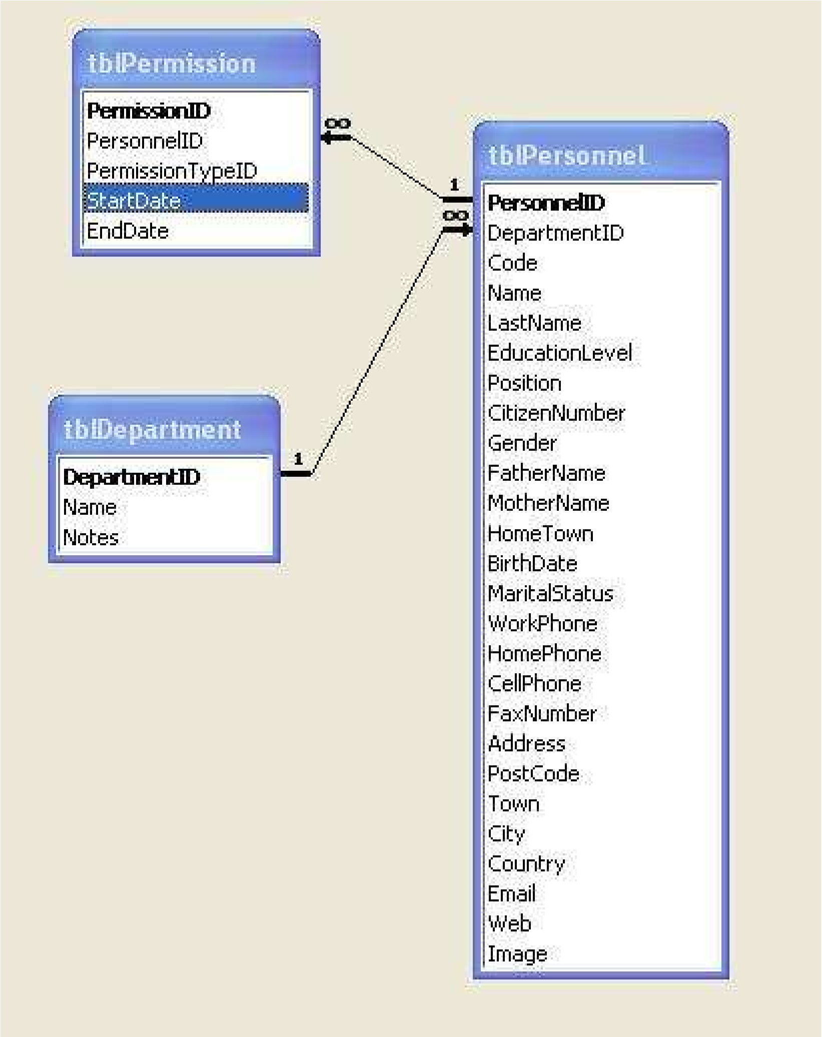
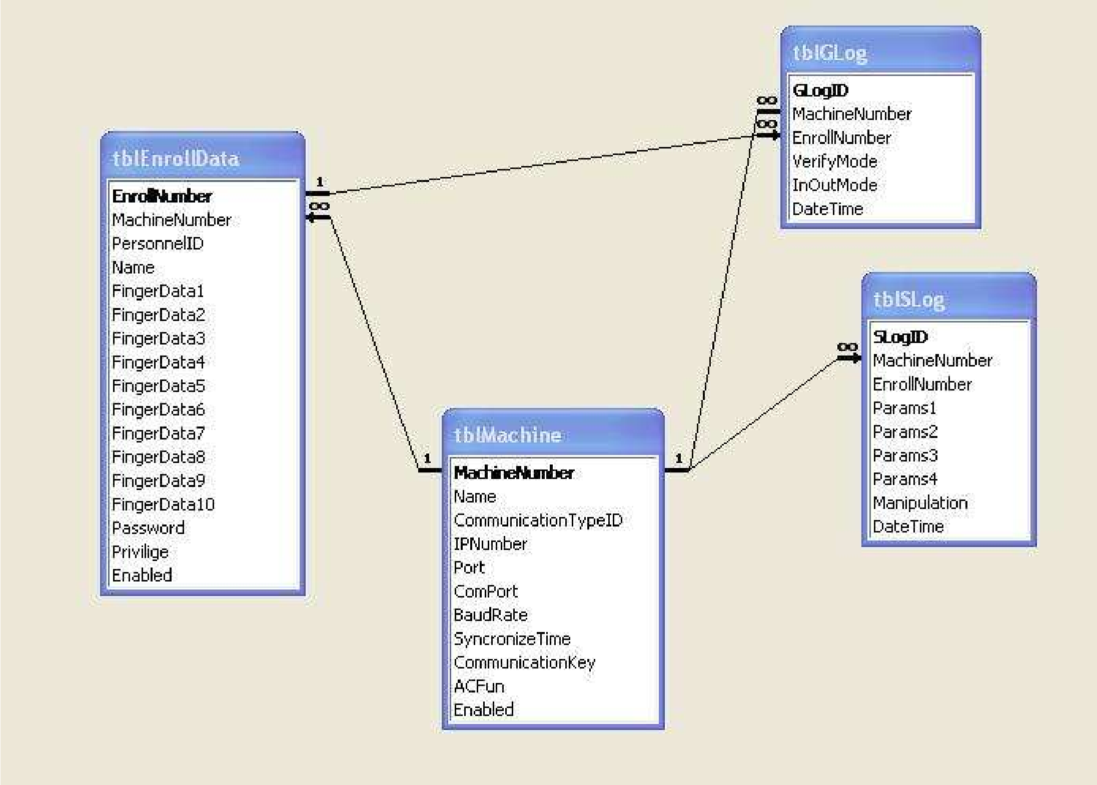


Figure B.1 Relation between tables



tblEnrollData

EnrollNumber
MachineNumber
PersonnelID
Name
FingerData1
FingerData2
FingerData3
FingerData4
FingerData5
FingerData6
FingerData7
FingerData8
FingerData9
FingerData10
Password
Privilige
Enabled

tblMachine

MachineNumber
Name
CommunicationTypeID
IPNumber
Port
ComPort
BaudRate
SyncronizeTime
CommunicationKey
ACFun
Enabled

tblGLog

GLogID
MachineNumber
EnrollNumber
VerifyMode
InOutMode
DateTime

tblSLog

SLogID
MachineNumber
EnrollNumber
Params1
Params2
Params3
Params4
Manipulation
DateTime

Figure B.2 Relation between tables.

APPENDIX C

FLOWCHART OF THE MAIN PROGRAM

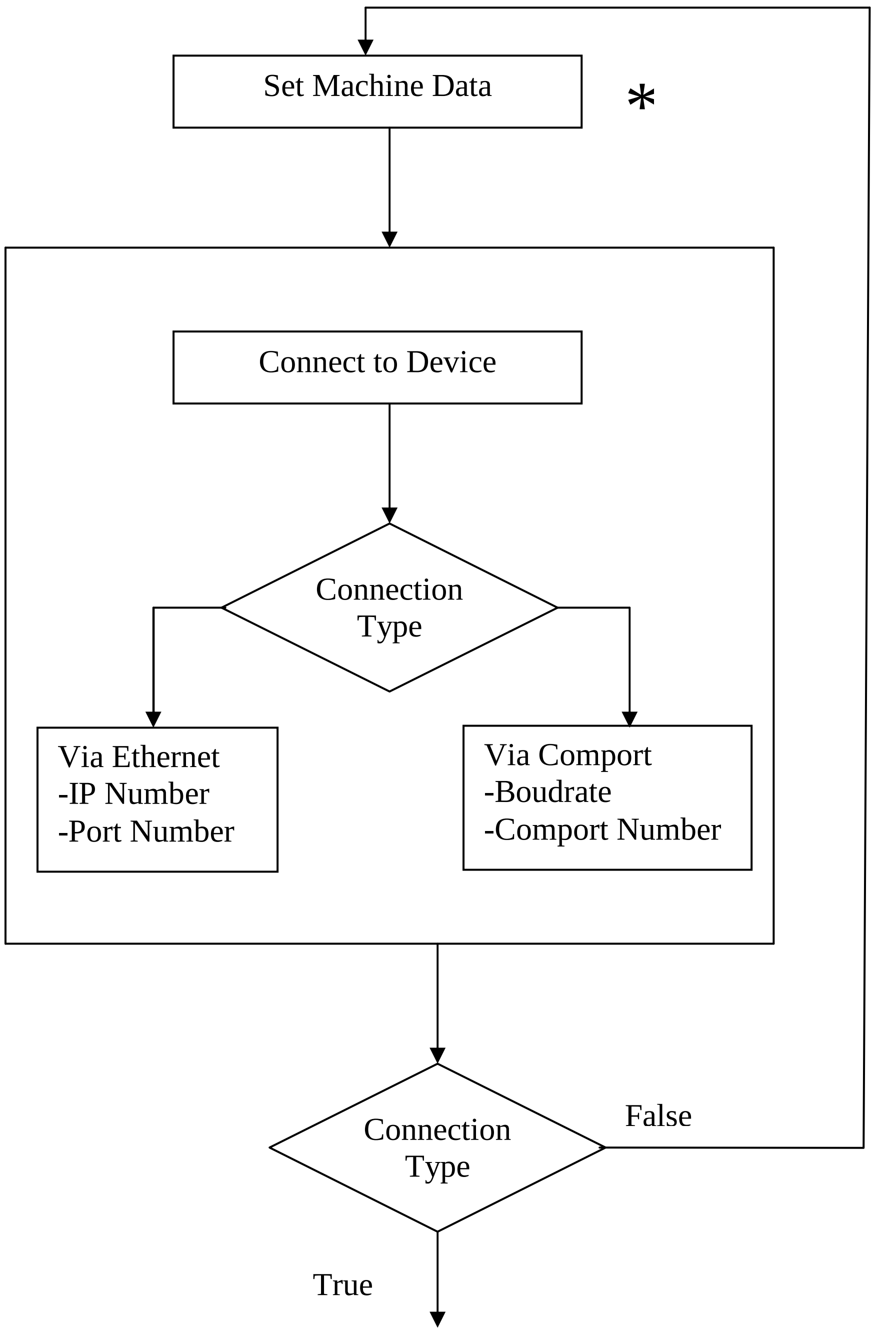


Figure C.1 Flowchart of machine connection procedure

* Set machine data includes ;

- Connection type
- Machine number
- IP number&Port number or Comport&Boudrate
- Connection type
- AC function time out (ms)

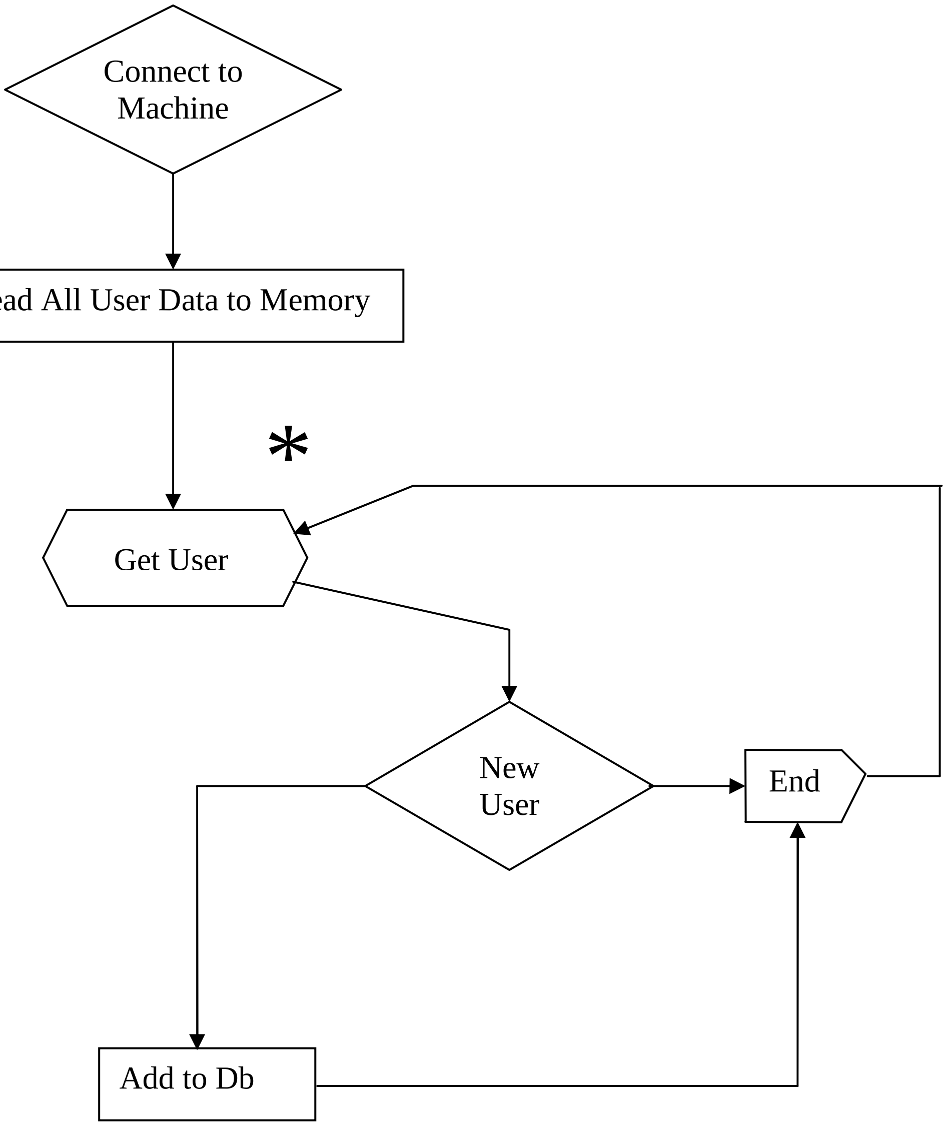


Figure C.2 Flowchart of getting user data

* Get user includes;

- Machine Number
- Enroll Number
- Password
- Name
- Privilige
- 10 finger data

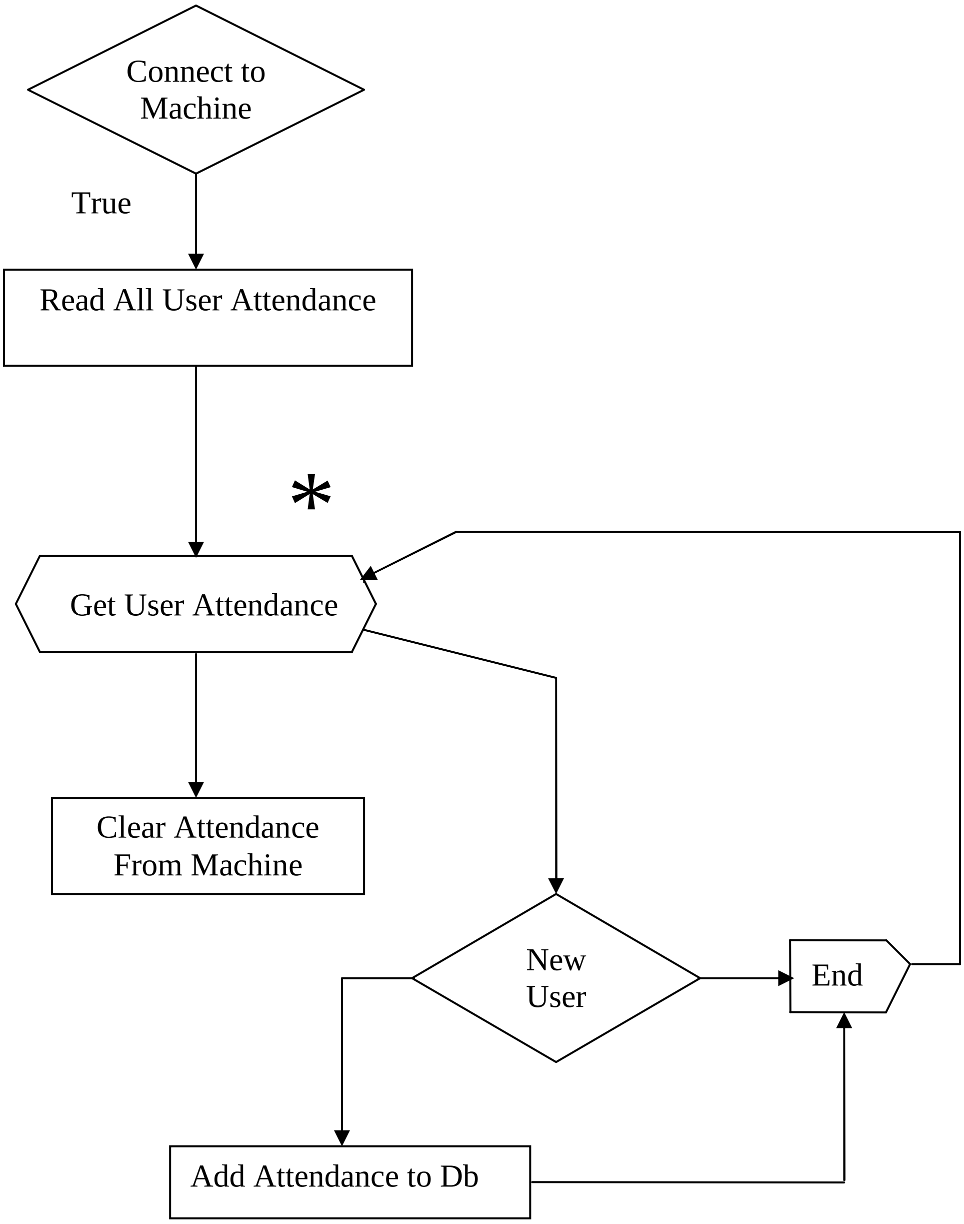


Figure C.3 Flowchart of getting attendance records

* Get user attendance includes;

- Date and Time
- In/Out
- Machine Number
- Enroll Number