



E-VOTING SYSTEM USING MULTIMODAL BIOMETRICS



PROJECT REPORT

Submitted by

MICHEAL THAINES MANISHA. S

Reg. No.:13BEC089

PAVITHRA.R

Reg. No.:13BEC102

RAJA RAJESHWARI. S

Reg. No.:13BEC117

In partial fulfilment for the award of the degree
of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE -641049

(An autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report “**E-VOTING USING MULTIMODAL BIOMETRICS**” is the bonafide work of **S. MICHEAL THAINES MANISHA [Reg. No: 13BEC089], R. PAVITHRA.R [13BEC102] and S. RAJA RAJESWARI [13BEC117]** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms . K. Jasmine M.E.,

PROJECT SUPERVISOR

Department of ECE

Kumaraguru College of Technology

Coimbatore-641049

SIGNATURE

Dr. K. Malarvizhi M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Department of ECE

Kumaraguru College of Technology

Coimbatore-641049

The candidates with Register No: 13BEC089, 13BEC102 and 13BEC117 are examined by us in the project viva-voce examination held on

.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our sincere thanks to the Management of Kumaraguru College of Technology and Joint Correspondent **Shri. Shankar Vanavarayar** for the kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr. R.S. Kumar, Ph. D.**, Kumaraguru College of Technology, who encouraged us at each and every step of the project.

We would like to thank **Dr. K. Malarvizhi, Ph.D.**, Head of the Department, Electronics and Communication Engineering, for her kind support and for providing necessary facilities to carry out the project work.

We wish to thank with everlasting gratitude to our Project Coordinator **Dr.A.Vasuki M.E., Ph.D.**, Department of Electronics and Communication Engineering for her consistent support throughout the course of this project work.

We are greatly privileged to express our deep sense of gratitude and heart felt thanks to our Project Guide **Ms. K. Jasmine, M.E.**, Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success and also we wish to convey our regards to all teaching and non-teaching staff of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support and abundant blessings in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

ABSTRACT

Electronic voting machine has already been developed and widely used in many developed countries use image processing technique to detect citizens. But only image processing is not enough. Keeping these problems in mind we have developed this project where raspberry pi will be used as host. But most of them use Radio Frequency ID. In developing countries RFID for each person does not exist. And using RFID is still a costly solution Hence we use Raspberry pi where camera will be used to take picture of citizen's national ID card and identify that this user is valid voter for that region. If the citizen is valid and also didn't vote then the person will be allowed to submit his/her vote. Each voting machine is locked by finger print access module. As the user is identified his/her finger print will be sent to a specific machine for voting. Each voting machine is networked with the central raspberry pi voting identification system. The IOT is regarded as the third wave of information technology after Internet and mobile communication network, which is characterized by more Recognition, more comprehensive interoperability and intelligence. It offers high security.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	Iv
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
2.1	An intra-image tracking algorithm for traffic recognition	2
2.2	Analysis of electronic voting System	2
3	PROJECT DESCRIPTION	3
3.1	Face Recognition	4
3.1.1	Steps Involved in Face Recognition	5
3.1.2	Weber's Law	5
3.2	Finger Vein Recognition	6
3.2.1	Steps Involved in Finger vein Recognition	6
4	EMBEDDED SYSTEMS	
4.1	Overview of embedded systems	8
4.2	Block diagram of an embedded system	10
4.3	Characteristics of embedded systems	11
4.3.1	Application Specific Systems	11

4.3.2	Reactive Systems	11
4.3.4	Heterogeneous Architectures	12
4.3.5	Harsh environment	12
4.3.6	Small and low weight	13
4.3.7	Cost sensitivity	13
4.3.8	Power management	13
5	BLOCK DIAGRAM	11
6	HARDWARE DESCRIPTION	15
6.1	Raspberry pi	15
6.1.1	Description	15
6.1.2	ARM vs x86	16
6.1.3	Windows vs Linux	18
6.2	Buzzer	19
6.2.1	Specifications	19
6.3	Camera	20
6.4	Motor	21
7	SOFTWARE DESCRIPTION	22
7.1	Python Language	22
7.2	Matlab	23
8	WORKING MODEL	24
9	MODULE SUMMARY	25
9.1	Steps for creating email using Rapberry pi	25
9.2	Steps for send way2sms on Rapberry pi	26

	9.2.1	SMS Through way2SMS	26
	9.2.2	Working	26
10		CONCLUSION	27
11	FUTURE	SCOPE	28
12	APPENDIX		28
13	REFERENCE		35

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Face Recognition	4
1.2	Finer Vein	7
1.3	Block diagram of embedded system	10
1.4	Block diagram of e-voting system	14
1.5	Raspberry Pi B+	16
1.6	Buzzer	19
1.7	Camera	20
1.8	Motor	21
1.9	Working Model	24
1.10	Mail Output	25
1.11	SMS Output	27

ABBREVIATIONS

Rpi	Raspberry Pi
RISC	Reduced Instruction Set Computer
LAN	Local Area Network
IoT	Internet of Things
CISC	Complex Instruction Set Computer
ARM	Advanced risk machine
GSM	Global System for mobile communication
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver /Transmitter

CHAPTER 1

1.INTRODCUCTION :

An electronic voting (e-voting) system is a voting system in which the election data is recorded, stored and processed primarily as digital information. It uses an electronic means of casting and counting votes. Some of these advantages are lesser cost, faster tabulation of results, greater accuracy, and lower risk of human and mechanical errors. It offers improved accessibility for the people with disabilities, and it provides multiple-language support for the ballots .Electronic voting is gaining in popularity around the world. Electronic voting is a term that may encompass several different types of voting, embracing both electronic means of casting a vote and electronic means of counting votes. A reliable cost effective secure electronic voting system that can be used in cost effectively way. The important obstacle in any e-voting system across the world is the security issue. Election's results may be modified when delivered to the Higher Elections Committee, unauthorized voter may vote instead of the eligible voter, a vote may not be calculated; also the voter has to ensure that nobody has the possibility to know his ballot data. The proposed Voting Model System overcomes these obstacles. Security evaluation experiments are performed successfully to the proposed system proving that it satisfies privacy, accuracy, reusability, eligibility and integrity.

CHAPTER 2

2. LITERATURE SURVEY

2.1 An intra-image tracking algorithm for traffic sign recognition

A detection subsystem which includes several segmentations that can work in parallel for each image. However, even though a sign can be detected several times, the final result at the output system must give a single stroke for each sign. For this purpose, after the recognition process, the intra-image tracking unifies detections and establishes correspondences according to criteria based on color, geometric shape and position. For each incoming image a new track is initiated and all identified signs are compared, existing a structure that contains the tracked objects. If the recognition results are not the same for the different detections of a sign, the tracking algorithm decision is given by majoritary vote.

2.2 Analysis of an electronic voting system

A security analysis of the source code to one such machine used in a significant share of the market. Our analysis shows that this voting system is far below even the most minimal security standards applicable in other contexts. We identify several problems including unauthorized privilege escalation, incorrect use of cryptography, vulnerabilities to network threats, and poor software development processes. Any paperless electronic voting system might suffer similar flaws, despite any certification it could have otherwise received. We suggest that the best solutions are voting systems having a voter-verifiable audit trail, where a computerized voting system might print a paper ballot that can be read and verified by the voter.

CHAPTER 3

3 .PROJECT DESCRIPTION:

The main aim of the project is to provide high security features in voting system. We have used two features for improving the security .

They are :

3.1 Face recognition

3.2 Finger vein recognition

A Face recognition system is a computer application capable of identifying or verifying a person from a digital image. Hence this is done by comparing selected facial features from the image. In this project the face is recognized with the help of camera module using MATLAB software. The image is trained in the database and the real time image which is captured using the camera module is compared with the database for authentication using Weber's Law. Once the face is authenticated, the finger vein window is opened. The finger vein images are compared with the database and verified for authentication. If the authentication fails:

- Buzzer will ring to intimate the people.
- SMS will be sent to the registered number
- E mail will be sent to the registered id.

3.1 FACE RECOGNITION

Face recognition is a biometric which uses computer software to determine the identity of the individual. Face recognition falls into the category of biometrics which is “the automatic recognition of a person using distinguishing traits”. Other types of biometrics include fingerprinting, retina scans, and iris scan.

The eigenface face recognition system can be divided into two main segments: creation of the eigenface basis and recognition, or detection, of a new face. The eigenface technique is simple, efficient, and yields generally good results in controlled circumstances. The system was even tested to track faces on film.

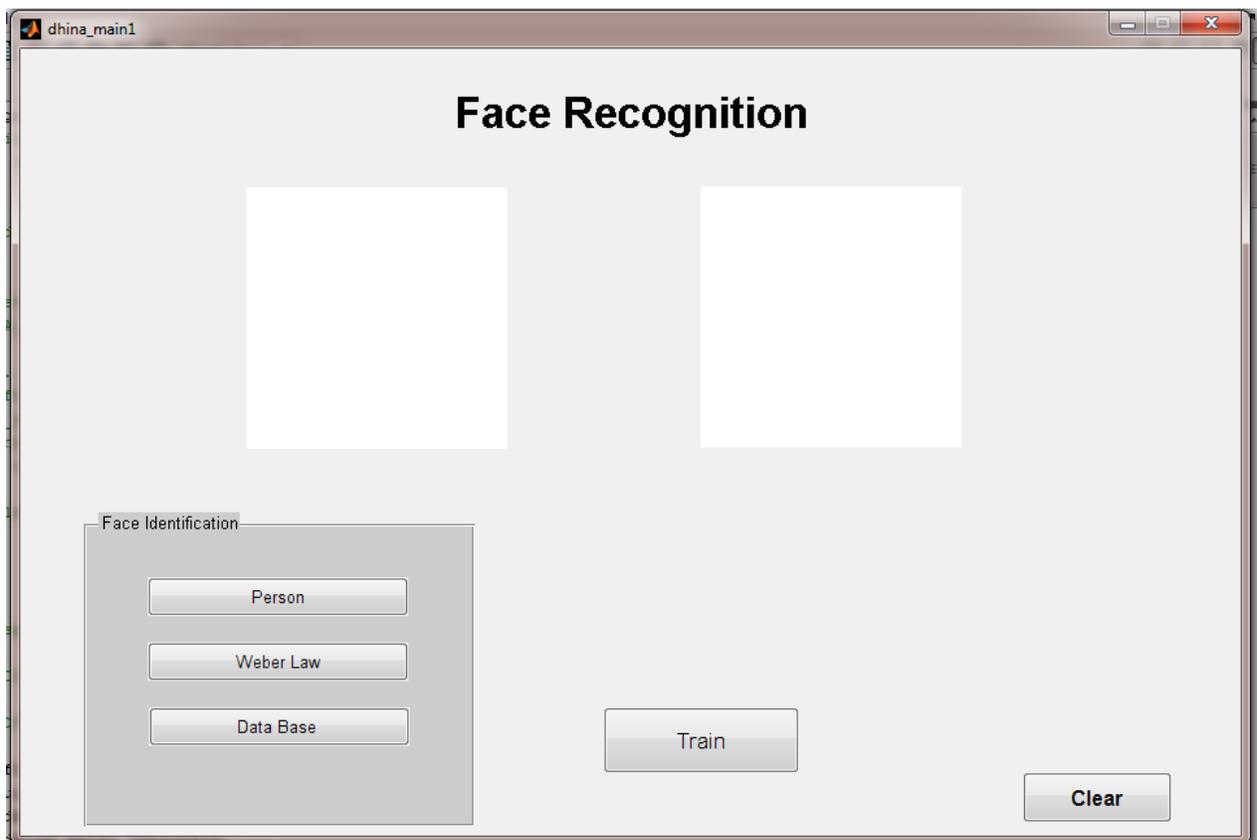


FIG 1.1 FACE RECOGNITION

3.1.1 STEPS INVOLVED IN FACE RECOGNITION:

- Open the face recognition window.
- Train the user image in the database.
- Capture the user image in real time.
- Apply the Weber's Law.
- If the database image and the real time image matches then the face is authenticated , else unauthenticated

3.1.2 WEBERS LAW:

Webers law states that "the ratio of the increment threshold to the background intensity is a constant". So when you are in a noisy environment you must shout to be heard while a whisper works in a quiet room. And when you measure increment thresholds on various intensity backgrounds, the thresholds increase in proportion to the background.

$$k = \frac{\Delta I}{I}$$

Where,

Delta I => the difference threshold
I => initial stimulus intensity
k=> constant

3.2 FINGER VEIN RECOGNITION:

Finger vein authentication is a method that specifies the individual using the vein pattern inside one's finger. During authentication the finger vein image is captured and compared against the stored template of the user. Finger vein recognition is a method of biometric authentication that uses pattern-recognition techniques based on images of human finger vein patterns beneath the skin's surface. Finger vein recognition is one of many forms of biometrics used to identify individuals.

ADVANTAGES :

1. The accuracy level of finger vein is very high
2. The security level is high
3. It offers long term stability
4. The size of the template is small

3.2.1 STEPS INVOLVED IN FINGER VEIN RECOGNITION:

- The finger vein images are loaded in the database.
- The finger vein input is selected
- Both the images are compared.
- If the images are matched then the person is allowed to vote and verify their identity

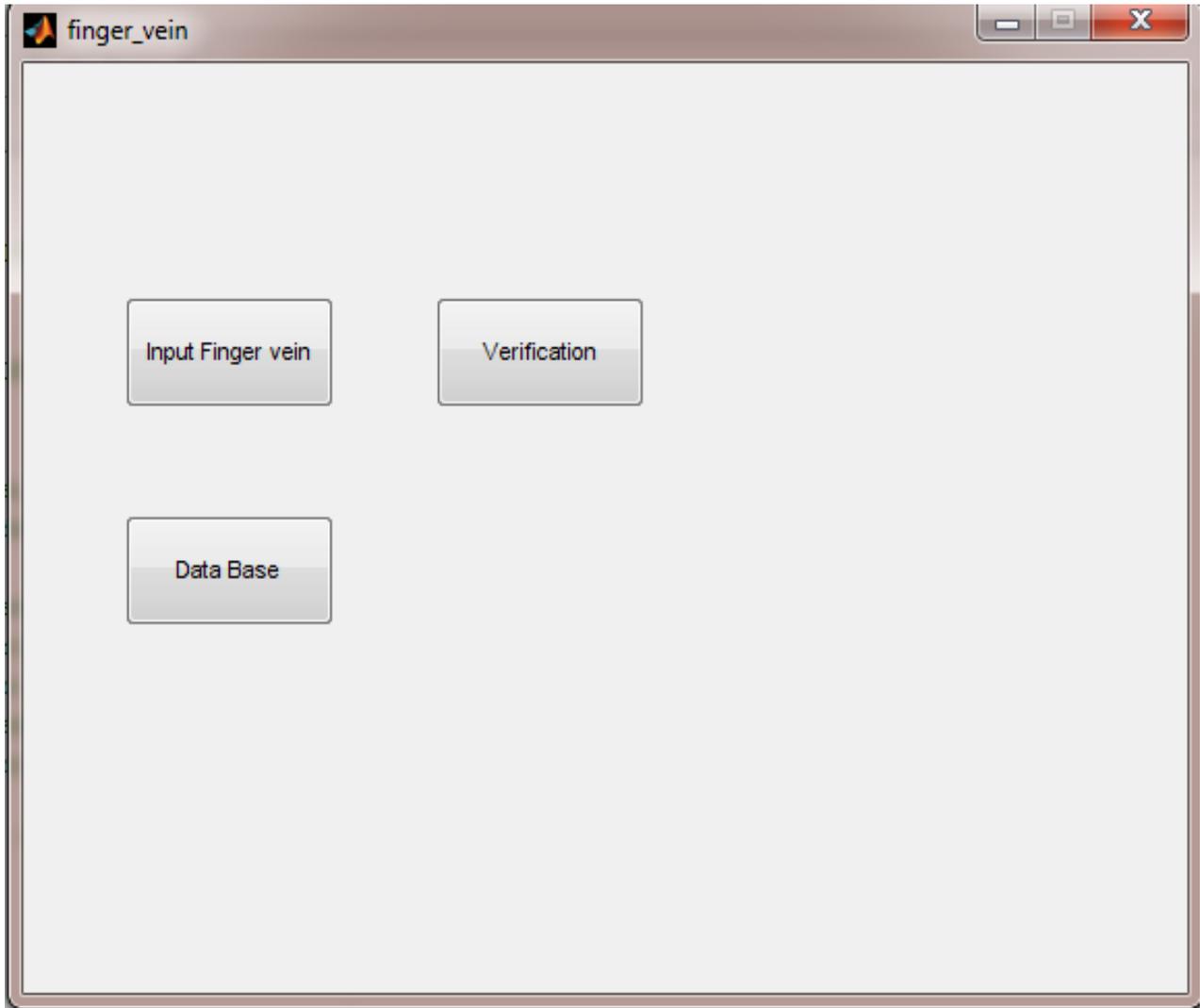


FIG 1.2 FINGER VEIN RECOGNITION

CHAPTER 4

4. EMBEDDED SYSTEMS

4.1 Overview of embedded systems

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems — such as the operating systems and microprocessors which power them — but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

Embedded systems provide several functions

- Monitor the environment; embedded systems read data from input sensors. This data is then processed and the results displayed in some format to a user or users
- Control the environment; embedded systems generate and transmit commands for actuators.
- Transform the information; embedded systems transform the data collected in some meaningful way, such as data compression /decompression

Although interaction with the external world via sensors and actuators is an important aspect of embedded systems, these systems also provide functionality specific to their applications. Embedded systems typically execute applications such as control laws, finite state machines, and signal processing algorithms. These systems must also detect and react to faults in both the internal computing environment as well as the surrounding electromechanical systems.

There are many categories of embedded systems, from communication devices to home appliances to control systems. Examples include;

- Communication devices
e.g.: modems, cellular phones
- Home Appliances
e.g.: CD player, VCR, microwave oven
- Control Systems
e.g.: Automobile anti-lock braking systems, robotics, satellite control

4.2 Block diagram of an embedded system:

An embedded system usually contains an embedded processor. Many appliances that have a digital interface -- microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an operating system. Others are very specialized resulting in the entire logic being implemented as a single program. These systems are embedded into some device for some specific purpose other than to provide general purpose computing . A typical embedded system is shown in Fig 1.3

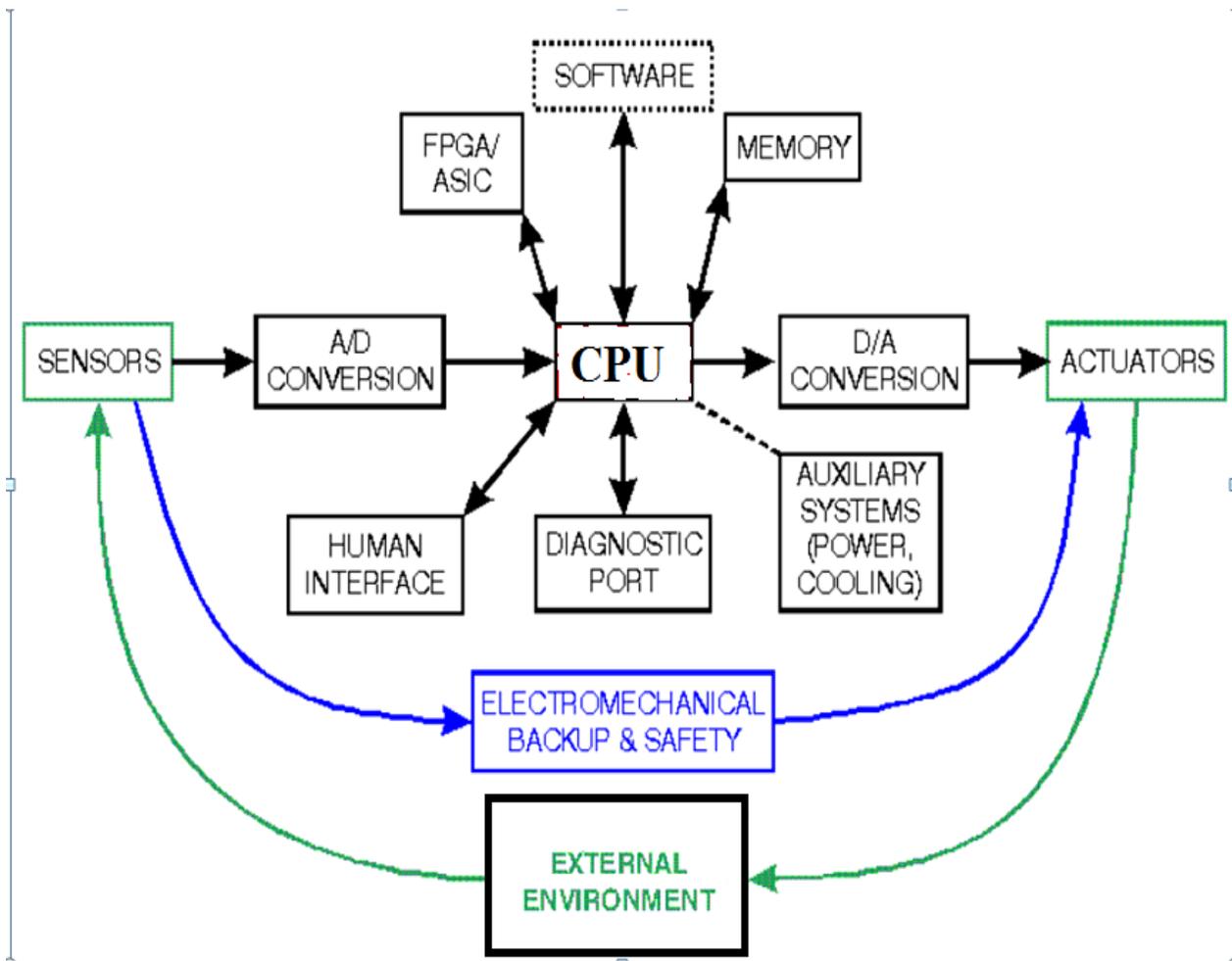


FIG 1.3 BLOCK DIAGRAM OF EMBEDDED SYSTEM

4.3 Characteristics of embedded systems

Embedded systems are characterized by a unique set of characteristics. Each of these characteristics imposed a specific set of design constraints on embedded systems designers. The challenge to designing embedded systems is to conform to the specific set of constraints for the application.

4.3.1 Application Specific Systems:

Embedded systems are not general-purpose computers. Embedded system designs are optimized for a specific application. Many of the job characteristics are known before the hardware is designed. This allows the designer to focus on the specific design constraints of a well-defined application. As such, there is limited user reprogrammability. Some embedded systems, however, require the flexibility of reprogrammability. Programmable DSPs are common for such applications.

4.3.2 Reactive Systems

As mentioned earlier, a typical embedded systems model responds to the environment via sensors and control the environment using actuators. This requires embedded systems to run at the speed of the environment. This characteristic of embedded system is called “reactive”. Reactive computation means that the system executes in response to external events. External events can be either periodic or aperiodic. Periodic events make it easier to schedule processing to guarantee performance.. The maximum event arrival rate must be estimated in order to accommodate worst case situations. Real time system operation means that the correctness of a computation depends, in part, on the time at which it is delivered. Systems with this requirement must often design to worst case performance.

4.3.3 Distributed Systems

A common characteristic of an embedded system is one that consists of communicating processes executing on several CPUs or ASICs which are connected by communication links. The reason for this is economy. Economical 4 8-bit microcontrollers may be cheaper than a 32-bit processors. Even after adding the cost of the communication links, this approach may be preferable. In this approach, multiple processors are usually required to handle multiple time-critical tasks. Devices under control of embedded systems may also be physically distributed.

4.3.4 Heterogeneous Architectures

Embedded systems often are composed of heterogeneous architectures (Fig 1.2). They may contain different processors in the same system solution. They may also be mixed signal systems. The combination of I/O interfaces, local and remote memories, and sensors and actuators makes embedded system design truly unique. Embedded systems also have tight design constraints, and heterogeneity provides better design flexibility.

4.3.5 Harsh environment

Many embedded systems do not operate in a controlled environment. Excessive heat is often a problem, especially in applications involving combustion (e.g., many transportation applications). Additional problems can be caused for embedded computing by a need for protection from vibration, shock, lightning, power supply fluctuations, water, corrosion, fire, and general physical abuse.

4.3.6 Small and low weight

Many embedded computers are physically located within some larger system. The form factor for the embedded system may be dictated by aesthetics. For example, the form factor for a missile may have to fit inside the nose of the missile. One of the challenges for embedded systems designers is to develop non-rectangular geometries for certain solutions. Weight can also be a critical constraint. Embedded automobile control systems, for example, must be light weight for fuel economy. Portable CD players must be light weight for portability purposes.

4.3.7 Cost sensitivity

Cost is an issue in most systems, but the sensitivity to cost changes can vary dramatically in embedded systems. This is mainly due to the effect of computer costs have on profitability and is more a function of the proportion of cost changes compared to the total system cost.

4.3.8 Power management

Embedded systems have strict constraints on power. Given the portability requirements of many embedded systems, the need to conserve power is important to maintain battery life as long as possible. Minimization of heat production is another obvious concern for embedded systems.

CHAPTER 5

5.BLOCK DIAGRAM:

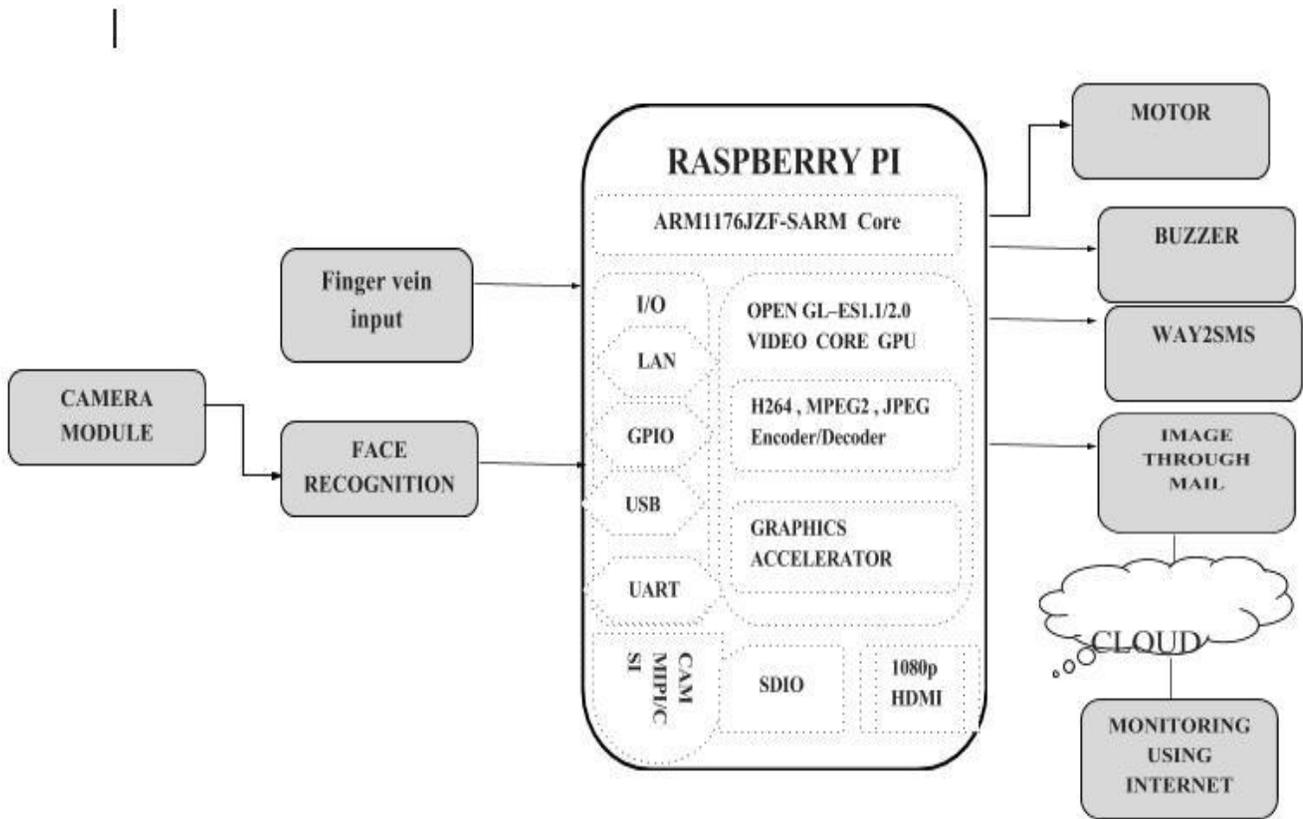


FIG 1.4 BLOCK DIAGRAM OF E-VOTING SYSTEM

CHAPTER 6

6. HARDWARE DESCRIPTION:

Components Required:

- ARM 11
- RS232 Serial CommunicationKeyboard
- Raspberry pi
- Buzzer
- Camera
- Motor

6.1. Raspberry Pi:

6.1.1 Description:

Raspberry Pi board is a miniature marvel, packing considerable computing power into a footprint no larger than a credit card. It's capable of some amazing things, but there are a few things you're going to need to know before you plunge head-first into the bramble patch.

Raspberry Pi B+:

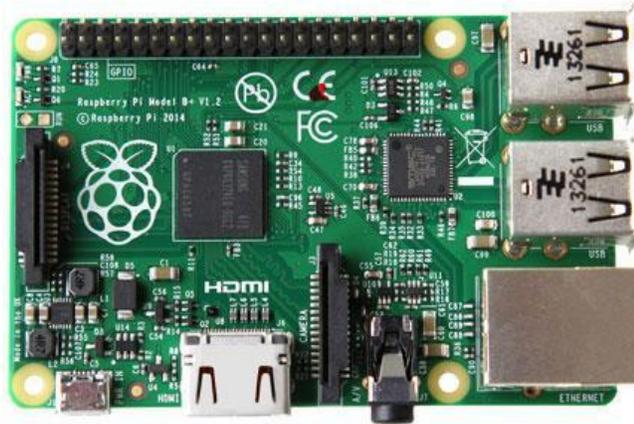


FIG 1.5 RASPBERRY PI B+

6.1.2 ARM vs. x86

The heart of the Raspberry Pi system is a Broadcom BCM2837 system-on-chip (SoC) multimedia processor. This means that the vast majority of the system's components, including its central and graphics processing units along with the audio and communications hardware, are built onto that single component hidden beneath the 256 MB memory chip at the centre of the board. It's not just this SoC design that makes the BCM2837 different to the processor found in your desktop or laptop, however it also uses a different instruction set architecture (ISA), known as ARM.

The BCM2837 SoC, located beneath a Hynix memory chip Developed by Acorn Computers back in the late 1980s, the ARM architecture is a relatively uncommon sight in the desktop world. Where it excels, however, is in mobile devices: the phone in your pocket almost certainly has at least one ARM-based processing core hidden away inside. Its combination of a simple reduced instruction set (RISC) architecture and low power draw make it the perfect choice over desktop chips with high power demands and complex instruction set (CISC) architectures. The ARM-based BCM2837 is the secret of how the Raspberry Pi is able to operate on just the 5V 1A power supply provided by the onboard micro-USB port. It's also the reason why you won't find any heat-sinks on the device: the chip's low power draw directly translates into very little waste heat, even during complicated processing tasks. It does, however, mean that the Raspberry Pi isn't compatible with traditional PC software. The majority of software for desktops and laptops is built with the x86 instruction set architecture in mind, as found in processors from the likes of AMD, Intel and VIA. As a result, it won't run on the ARM-based Raspberry Pi. The BCM2837 uses a generation of ARM's processor design known as ARM11, which in turn is designed around a version of the instruction set architecture known as ARMv6. This is worth remembering: ARMv6 is a lightweight and powerful architecture, but has a rival in the more advanced ARMv7 architecture used by the ARM Cortex family of processors. Software developed for ARMv7, like software developed for x86, is sadly not compatible with the Raspberry Pi's BCM2837 although developers can usually convert the software to make it suitable.

6.1.3 Windows vs. Linux

Another important difference between the Raspberry Pi and your desktop or laptop, other than the size and price, is the operating system—the software that allows you to control the computer.

The majority of desktop and laptop computers available today run one of two operating systems: Microsoft Windows or Apple OS X. Both platforms are closed source, created in a secretive environment using proprietary techniques. These operating systems are known as closed source for the nature of their source code, the computer-language recipe that tells the system what to do. In closed-source software, this recipe is kept a closely-guarded secret. Users are able to obtain the finished software, but never to see how it's made.

The Raspberry Pi, by contrast, is designed to run an operating system called GNU/Linux—hereafter referred to simply as Linux. Unlike Windows or OS X, Linux is open source: it's possible to download the source code for the entire operating system and make whatever changes you desire. Nothing is hidden, and all changes are made in full view of the public. This open source development ethos has allowed Linux to be quickly altered to run on the Raspberry Pi, a process known as porting. At the time of this writing, several versions of Linux—known as distributions—have been ported to the Raspberry Pi's BCM2837 chip, including Debian, Fedora Remix and Arch Linux. The different distributions cater to different needs, but they all have something in common: they're all open source. They're also all, by and large, compatible with each other: software written on a Debian system will operate perfectly well on Arch Linux and vice versa. Linux isn't exclusive to the Raspberry Pi. Hundreds of different distributions are available for desktops, laptops and even mobile devices; and Google's popular Android platform is developed on top of a Linux core.

6.2 BUZZER:



FIG 1.6 BUZZER

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sens expansion module and the board in combination, can complete a simple circuit design, to "plug and play".

6.2.1 Specifications:

- On-board passive buzzer
- On-board 8550 triode drive
- Can control with single-chip microcontroller IO directly
- Working voltage: 5V

6.3 CAMERA:



FIG 1.7 CAMERA

Resolution : 320 x 240, 640 x 480

Frame Rate (max) : 30fps

The real time images are captured by the camera to detect the presence of human in the room eg : real time web camera. The first 5 frames are taken as the reference frame and that image is stored and then by starting the camera it will start to capture the images. The images will be recorded in the software tool . The coding is done such that , in a time interval of 1 ms every frame is taken and subtracted with the reference frame. That is vector subtraction is performed between the two frames .If there is any variations or difference occurs between the two frames then it is an indication that the human is present .This is called image processing. The camera records the voters image and it is used to compare with the registered image. Once the image comparison fails buzzer will ring. The camera resolution should be good in order to capture the image efficiently.

6.4 MOTOR:



FIG 1.8 9V DC MOTOR

A stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical movements. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. The motor's rotation has several direct relationships to these applied input pulses. The sequence of the applied pulses is directly related to the direction of motor shafts rotation. The speed of the motor shafts rotation is directly related to the frequency of the input pulses and the length of rotation is directly related to the number of input pulses applied. The motor used is 9v DC motor.

CHAPTER 7

7.SOFTWARE DESCRIPTION:

- Raspbi Jessi(OS)
- Language-Python surveillance.
- MATLAB

7.1 PYTHON LANGUAGE:

Assuming that you have experience with programming in some type-safe language, aims at upgrading your knowledge about scripting, focusing on the Python language. Python has many attractive features that in my view makes it stand out from other dynamically typed languages:

– Python is easy to learn because of the very clean syntax, extensive built-in runtime checks help to detect bugs and decrease development time, programming with nested, heterogeneous data structures is easy, object-oriented programming is very convenient, there is support for efficient numerical computing, and the integration of Python with C, C++, Fortran, and Java is very well supported. If you come from FORTRAN, C, C++, or Java, you will probably find the following features of scripting with Python particularly advantageous:

Since the type of variables and function arguments are not explicitly written, a code segment has a larger application area and a better potential for reuse.

1. There is no need to administer dynamic memory: just create variables when needed, and Python will destroy them automatically.
2. Keyword arguments give increased call flexibility and help to document the code.

3. The ease of setting up and working with arbitrarily nested, heterogeneous lists and dictionaries often avoids the need to write your own classes to represent non-trivial data structures.

4. Any Python data structure can be dumped to the screen or to file with a single command, a highly convenient feature for debugging or saving data between executions.

7.2 MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB coding is used for face recognition and finger vein recognition. Library functions are being accessed for the recognitions.

Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building.

CHAPTER 8

8. WORKING MODEL :

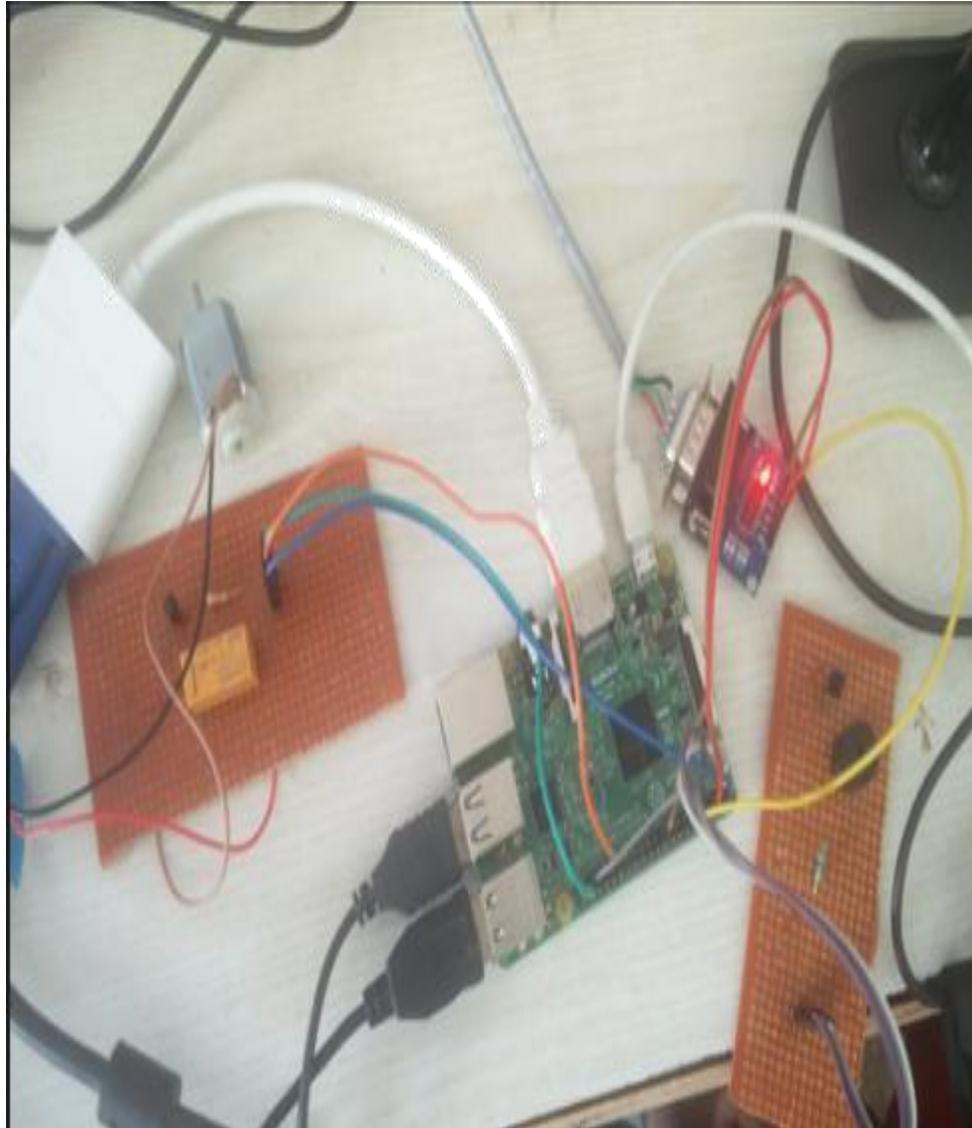


FIG 1.9 WORKING MODEL

CHAPTER 9

9. MODULE SUMMERY:

9.1 Steps to Creating email using Raspberry Pi:

- Step-by-Step Illustrated guide to sending emails through the command line in linux .

Setup your Raspberry Pi.

- Setup a Gmail Account to work with smtp.
- Install and configure the smtp program.
- Secure the smtp configuration file.
- Send an email from the command line

MAIL OUTPUT:

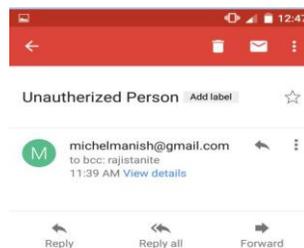


FIG 1.10 MAIL OUTPUT

9.2 Steps to send way2sms using Raspberry Pi:

9.2.1 Sending SMS through Way2SMS:

The parameters required to send the SMS are

- 1.The account must be created in way2 SMS application.
- 2.The sender and reciever contact number must be specified in the program.

9.2.2 Working:

When a person is detected in a room, a signal is sent to the controller, which initiates through SMS.

By sending an SMS to yourself with the conditions of your choice, as coded in your python script, you can create completely customized alerts. There's a native library in Python to send SMS `smtplib`, No need to install external libraries. If you want to login for an account, you will need an account in way to SMS so you can be safely identified and contacted. The parameters for the Way2SMS server. First, The server location (or its IP address), then the port to use

To have sent a message with an Way2SMS need to indicate:

- a. The user name
- b. Password

SMS OUTOUT :

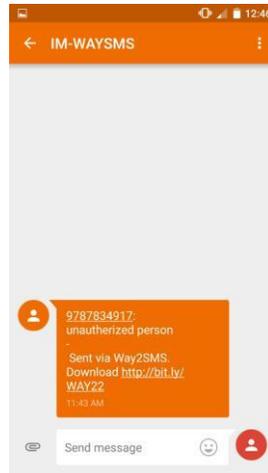


FIG 1.11 SMS OUTPUT

10.CONCLUSION :

This project is designed using Raspberry Pi microcontroller. The proposed e-voting system is designed especially to solve the cost effective, accuracy and transparency problems in a highly secured approach. It offers a systematic way of voting. Identification of appropriate technology and its extent to use for secure electronic voting is the prime concern in electoral process. It is achieved using this project. This project is designed in such a way that it offers very high security.

11.FUTURE SCOPE :

Reliable and secure electronic voting system presented in this project is promising; still there is scope for improvement in the proposed framework. In future, this framework can be updated to support Internet voting. Malpractices while voting can be controlled to a large extent.

12.APPENDIX:

12.1 FACE RECOGNITION:

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @dhina_main1_OpeningFcn, ...
                  'gui_OutputFcn',  @dhina_main1_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

end

function dhina_main1_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
a = ones(256,256);
axes(handles.axes1);
imshow(a);
axes(handles.axes2);
imshow(a);
guidata(hObject, handles);

end

function varargout = dhina_main1_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

end

function pushbutton1_Callback(hObject, eventdata, handles)
```

```

imaqhwinfo;
    info = imaqhwinfo('winvideo',1);
    celldisp(info.SupportedFormats)
    vid=videoinput('winvideo',1,'YUY2_640x480');
    set(vid, 'ReturnedColorSpace', 'rgb');

    preview(vid);
    pause;
    data=getsnapshot(vid);

    axes(handles.axes1);
imshow(data);
    X = data;
    im =X;

    data = imresize(data,[128 128]);
    imwrite(data, '50.jpg');
imwrite(data, 'hi.jpg')

if isrgb(X)
    X = rgb2gray(X);
    X = facedetec(X);
    X = X;
    X2 =uint8(X);
    X2 = histeq(X2);
end

closepreview(vid)
handles.X = X;
handles.X2 = X2;
guidata(hObject, handles);
end

function pushbutton2_Callback(hObject, eventdata, handles)
X = handles.X2;
imshow(X);

nn =9;
    alfa =2;
    sigma =1;
    y1 = weberlaw(nn, alfa, sigma, X);
axes(handles.axes2);
    imshow(y1);
handles.y1 = y1;
guidata(hObject, handles);
end
function pushbutton3_Callback(hObject, eventdata, handles)
temp = [];
y1 = handles.y1;
features12 = [];

sample =10;

```

```

for k = 1 : 1
    k1 = int2str(k);
    k2 = strcat(k1, '.bmp');

I = imread(k2);

nn =9;
    alfa =2;
    sigma =1;
    I = histeq(I);
y = weberlaw(nn, alfa, sigma, I);
axes(handles.axes2);
imshow(y);
I = y1;

figure
imshow(I);

T = y;

figure
imshow(T);

[r1 c1] = size(I);

image22=T-mean(mean(T));
[r c] = size(T);

for i=1:1:r1-r+1
    for j=1:1:c1-c+1
        f = I(i:i+r-1,j:j+c-1,:);
        t = T(:, :, :);
        C = corr2(t, f);

        f = max(max(C));

        dd(i,j) = f;

    end
end
ddl = max(max(dd));
mm = 0.1
if ddl > mm
    warndlg('Autherised');
    open ('finger_vein.m');
    open ('finger_vein.fig');

else
    warndlg(' Not Autherised');
s = 'B'

s1 = serial('COM1', 'BaudRate', 9600);
fopen(s1);
fprintf(s1,s);
fclose(s1);
delete(s1);

```

```

        end
        handles.dd1 = dd1;
handles.mm = mm;
mk = imread('db.jpg');
axes(handles.axes2);
imshow(mk);
end

guidata(hObject, handles);

end

function pushbutton7_Callback(hObject, eventdata, handles)

clc
close all

a = ones(256,256);
axes(handles.axes1);
imshow(a);
axes(handles.axes2);
imshow(a);
set(handles.t4, 'string', '');

set(handles.t5, 'string', '');
clear all
end

function pushbutton6_Callback(hObject, eventdata, handles)

dd1 = handles.dd1;
mm = handles.mm;
if dd1 > mm

    set(handles.t4, 'string', 'Name');

    set(handles.t5, 'string', 'Pantech');
else

    set(handles.t4, 'string', 'Name');

    set(handles.t5, 'string', 'Non');

end

end

function pushbutton8_Callback(hObject, eventdata, handles)

clc
train
end

```

12.2 FINGER VEIN :

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @finger_vein_OpeningFcn, ...
                  'gui_OutputFcn',  @finger_vein_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function finger_vein_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

guidata(hObject, handles);

function varargout = finger_vein_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function inp_finger_vein_Callback(hObject, eventdata, handles)
cd test
file=uigetfile('*.jpg');
inp=imread(file);
cd ..
figure;imshow(inp);

handles.inp=inp;

guidata(hObject, handles);

function verification_Callback(hObject, eventdata, handles)

inp=handles.inp;
X1=mean(mean(inp));
load Z dbfeat;
qfeat=X1;

for i = 1:2
    ed = qfeat-dbfeat(i);
```

```

    edist(i) = abs(sum(ed))
end
k=min(edist);
if k < 0.47;
    msgbox('AUTHENTICATE');
    finger_vein
    s = 'A'

        s1 = serial('COM3','BaudRate',9600);
        fopen(s1);
        fprintf(s1,s);
        fclose(s1);
        delete(s1);

else
    errordlg('UN-AUTHENTICATE');
    s = 'B'

        s1 = serial('COM3','BaudRate',9600);
        fopen(s1);
        fprintf(s1,s);
        fclose(s1);
        delete(s1);
end

function data_base_Callback(hObject, eventdata, handles)
for dbi=1:2
    fname = strcat(int2str(dbi),'.jpg');
    cd Finger_VEIN_DB
        inp = imread(fname);
    cd ..
    inp = imresize(inp,[256,256]);

    if size(inp,3)>1
        inp = rgb2gray(inp);
    end
    Xfeat=mean(mean(inp));

    dbfeat(:,dbi)=[Xfeat];

end

save Z dbfeat;

```

PYTHON CODING :

```
import urllib2
import cookielib
from getpass import getpass
import sys
import os
from stat import *

message = raw_input("Enter text: ")
number = raw_input("Enter number: ")

if __name__ == "__main__":
    username = "your_login_id"
    passwd = "your_password"

    message = "+".join(message.split(' '))

#logging into the sms site
url = 'http://site24.way2sms.com/Login1.action?'
data = 'username='+username+'&password='+passwd+'&Submit=Sign+in'

#For cookies

cj= cookielib.CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))

#Adding header details
opener.addheaders=[('User-Agent','Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120')]
try:
    usock =opener.open(url, data)
except IOError:
    print "error"
    #return()

jession_id =str(cj).split('~')[1].split(' ')[0]
send_sms_url = 'http://site24.way2sms.com/smstoss.action?'
send_sms_data =
'ssaction=ss&Token='+jession_id+'&mobile='+number+'&message='+message+'&msgLen
=136'
opener.addheaders=[('Referer',
'http://site25.way2sms.com/sendSMS?Token='+jession_id)]
try:
    sms_sent_page = opener.open(send_sms_url,send_sms_data)
except IOError:
    print "error"
    #return()
print "success"
#return ()
```

REFERENCE :

- 2016 IEEE 6th International Conference on Advanced Computing.
- T. Kohno, A. Stubblefield, A. Rubin, and D. Wallach, “Analysis of an Electronic Voting System,” in *Proc. of IEEE Symp. Security and Privacy*, pp. 27-40, 2004.

