# FINGERPRINT BASED BIOMETRIC VOTING MACHINE

## Members involved:

1. MOGILI VISHAL GOUD – 17BCE7028
2. PEDARLA SAMPATH KUMAR – 17BCE7091
3. THOTA MANISH – 17BCN7041
4. JULURI KALYAN VARDHAN – 17BCD7012
5. DHARANIDARAN – 17BES7001
6. VUNNAM NAGASAI – 17BME7087

**Under the Guidance of**

## Prof. Vijaya,

Department of Computer Science and Technology,
VIT-AP,
Amaravati,
Andhra Pradesh – 522237.

## Summary of the Project:

Voting machines are the total combination of mechanical, electromechanical, or electronic equipment (including software, firmware, and documentation required to program control, and support equipment), that is used to define ballots; to cast and count votes; to report or display election results; and to maintain and produce any audit trail information. The first voting machines were mechanical but it is increasingly more common to use electronic voting machines.

A voting system includes the practices and associated documentation used to identify system components and versions of such components; to test the system during its development and maintenance; to maintain records of system errors or defects; to determine specific changes made after initial certification; and to make available any materials to the voter (such as notices, instructions, forms, or paper ballots).

Traditionally, a voting machine has been defined by the mechanism the system uses to cast votes and further categorized by the location where the system tabulates the votes.

Voting machines have different levels of usability, security, efficiency and accuracy. Certain systems may be more or less accessible to all voters, or not accessible to those voters with certain types of disabilities. They can also have an effect on the public's ability to oversee elections.

Electronic voting systems may offer advantages compared to other voting techniques. An electronic voting system can be involved in any one of a number of steps in the setup, distributing, voting, collecting, and counting of ballots, and thus may or may not introduce advantages into any of these steps.

Moreover it is also important that a false entry should not be made so for this one of the most secure methods for voting is using a biometric sensor like a fingerprint reader.

Fingerprints are one of many forms of biometrics used to identify individuals and verify their identity. Fingerprint recognition or fingerprint authentication refers to the automated method of verifying a match between two human fingerprints.

In this project we will be using a Fingerprint reader for providing access to the voter as well as making a  log if the person has voted or not

# Index:

## Introduction:

Biometrics is the science and technology of measuring and analyzing biological data. Biometrics refers to technologies that measure and analyze human body characteristics, such as DNA, fingerprints, eye retinas and irises, voice patterns, facial patterns and hand measurements, for authentication purposes.The field of biometrics was formed and has since expanded onto many types of physical identification. Among the several human fingerprints remain a very common identifier and the biometric method of choice among law enforcement. These concepts of human identification have lead to the development of fingerprint scanners that serve to quickly identify individuals and assign access privileges. The basic point of these devices is also to examine the fingerprint data of an individual and compare it to a database of other fingerprints.

In our project we have used fingerprint for the purpose of voter identification or authentication. As the thumb impression of every individual is unique, it helps in minimizing the error.A database is created containing the fingerprint images of all the voters as required. Illegal votes and repetition of votes is checked for in this system with accurate coding. Hence with the application of this fingerprint based EVM system elections could be made fair and free from rigging. Further that the elections would are no longer a tedious and expensive job.

Biometric Finger print devices are used in the Electronic Voting machine for voter verification. We have designed a finger print based voting machine where there is no need for the user to carry his ID which contains his required details. The person at the polling booth needs only to place his Finger on the device, thus allowing the acquisition of an on-spot fingerprint from the voter which serves as an identification. This Finger print reader reads the details from the tag. This data is passed onto the controlling unit for the verification. The controller fetches the data from the reader and compares this data with the already existing data stored during the registration of the voters. If the data matches with the pre-stored information of the registered fingerprint, the person is allowed to cast his vote. If not, a warning message is displayed on LCD and the person is barred from polling his vote. The vote casting mechanism is carried out manually using the push buttons. LCD is used to display the related messages, warnings and ensuing results

## Background:

This research was implemented using the Arduino. The system read the data from the Fingerprint module verify the data with the already stored data and take the next action. The system is totally designed using Arduino, Fingerprint module and pushbuttons. The Arduino is control by program using C/C++ to allow the interface with the Fingerprint Module, the Arduino controller verifies this data with the already existing data in the controller's memory and then implement the commands directed by the controller section.

Virendra Kumar Yadav et al. [5], an approach that will use the information provided by UIDAI in smart voting system. The proposed system procedure is carried out in mainly few stages: registration, verification and validation. These stages of proposed system are illustrated.

D. Ashok Kumar et al. [6] made a comparative Study on Fingerprint Matching Algorithms for EVM. Then fingerprint is match voter can vote to candidate by using EVM. Fingerprint is secure method for EVM.

Jefferson D., et al. [7] reviewed and computer of critique and security communication in secure voting system. The web based voting system being built by Accenture. And in security the fingerprint technology are uses.

Qijun Zhao, et al. [8] proposed an adaptive pore model for fingerprint pore extraction. Sweat pores have been recently employed for automated fingerprint recognition, in which the pores are usually extracted by using a computationally expensive skeletonization method or a unitary scale isotropic pore model.

R. Moheb et al. [9] proposed an approach to image extraction and accurate skin detection from web pages. Their system to extract images from web pages and then detect the skin color regions of these images.

Manvjeet Kaur et al. [10] proposed a fingerprint verification system using minutiae extraction technique. Most fingerprint recognition techniques are based on minutiae matching and have been well studied.

Hoi Le and The Duy Bui, [11] proposed online fingerprint identification with a fast and distortion tolerant hashing method. They present a specific contribution by introducing a new robust indexing scheme that is able not only to fasten the fingerprint recognition process but also improve the accuracy of the system.

## Problem Definition:

In 21st century society where electronic technology is growing at an ever increasing rate, it is difficult to understand why governments were not converting their paper based election systems to electronic form to guaranty "One Person – One Vote and to eliminate fraud and corruption.

An example of how a paper based voting system is with disabilities and vulnerable to corruption can be found in the elections, where the last election was invalidated due to fraudulent paper ballots used to stuff the ballot boxes and elect a president illegally. To repair this damage, it has already cost which could be a recurring cost if the fraud occurred again and it is difficult to bring charges against the people committing the crime due to lack of evidence and an audit trail that could be used as a "Chain of Evidence" by lawyers. Another example is when paper election ballots ran out at an American election and additional ballots were produced using a printer and make-shift process for creating the new ballots on white paper instead of the normal blue ballots. People rushed to obtain the new white ballots and quickly completed them and stuffed them into the ballot boxes in a manner that was not traceable and could have been fraudulently submitted, showing that even first world countries suffer from the use of paper based ballots.

## Objective:

The fingerprint voting project demands the user to submit Fingerprint at the polling booth. The project uses the Fingerprint technology and Arduino Systems to design this application. The main objective of this project is to design a system that asks to user to show his/her Fingerprint as an identity proof. The system reads the data from the Fingerprint and verifies the data which is already stored data in the database. If the given details match with the database data, the system allows the person to cast their vote. If the given Fingerprint data does not match with the stored data, the system immediately activates the display and the security authorities can come and take the further action

## Methodology/Procedure:

**Required Components:**

Arduino Uno

Finger Print Sensor Module

Push Buttons

LEDs -2

1K Resistor -3

2.2K resistor

Power

Connecting wires

Buzzer

16x2 LCD

Bread Board

**ESP8266 Wi-Fi Module:**

The ESP2866Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP2866 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP2866 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP2866 module is an extremely cost effective board with a huge, and ever growing, community. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP2866 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts. Wi-Fi Modules are shown below:

ESP-01 is the one of the most popular ESP8266 module available in the market. ESP8266 is a self contained SoC with integrated TCP/IP stack which helps any microcontroller having UART to access a wifi network. It can act as both WiFi access point as well as a WiFi client. It is pre-programmed with AT commands, so we can easily access and configure it using a microcontroller.

ESP8266 runs on 3.3V and its input pins are not 5V tolerant. So we need to reduce the 5V output of the Arduino Tx pin to 3.3V by using voltage dividing resistors to connect to Rx pin of ESP8266 module. Arduino TTL input pins will detect 3.3V as logic high, so we can directly connect 3.3V output of ESP8266 Tx to Arduino Rx pin.

**Arduino Uno:**

The Arduino Uno is a microcontroller board based on the ATmega328 as shown in Fig(iii). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.


The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

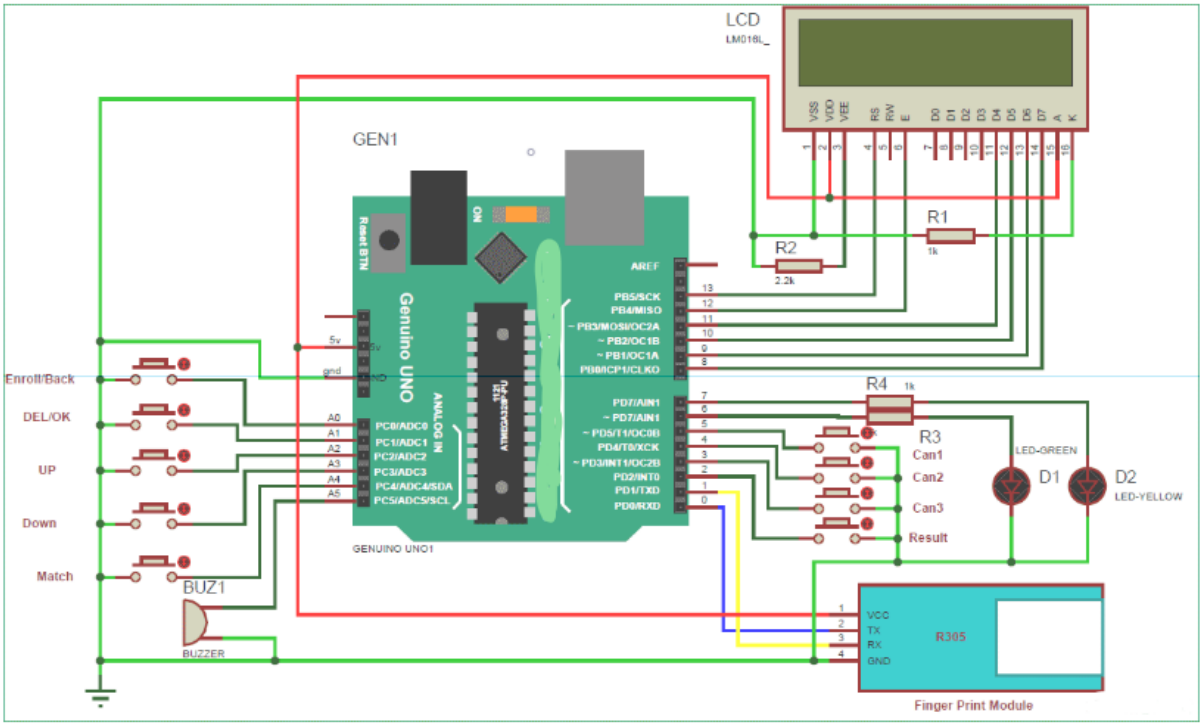
**Fingerprint Sensor Module**:

Finger Print Sensor Module or Finger Print Scanner is a module which captures finger's print image and then converts it into the equivalent template and saves them into its memory on selected ID (location) by Arduino. Here all the process is commanded by Arduino like taking an image of finger print, convert it into templates and storing location etc.

In this FingerPrint Voting Machine Circuit, we have used Finger Print Sensor Module to authenticate true voter by taking their finger input in the system. Here we are using 5 push buttons to Match, Enroll/back, Delete/OK, UP and Down. Enroll and Del key have double features here. Enroll key is used for enrolling new finger impression into the system and back

function as well. Means when the user wants to enroll new finger then he/she needs to press enroll key then LCD asks for the ID or Location where user wants to store the finger print output. Now if at this time user do not want to proceed further then he/she can press enroll key again to go back (this time enroll key behave as Back key). Means enroll key has both enrollment and back function. DEL/OK key also has same double function like when user enrolls new finger then he/she need to select finger ID or Location by using another two key namely UP AND DOWN now user needs to press DEL/OK key (this time this key behaves like OK) to proceed with selected ID or Location. Match key is used for whenever voter wants to vote then he/she needs to authenticate first for true voter by keeping finger on Finger Print Sensor, if he/she passed in this authentication then he/she can vote.

**Circuit Diagram:**

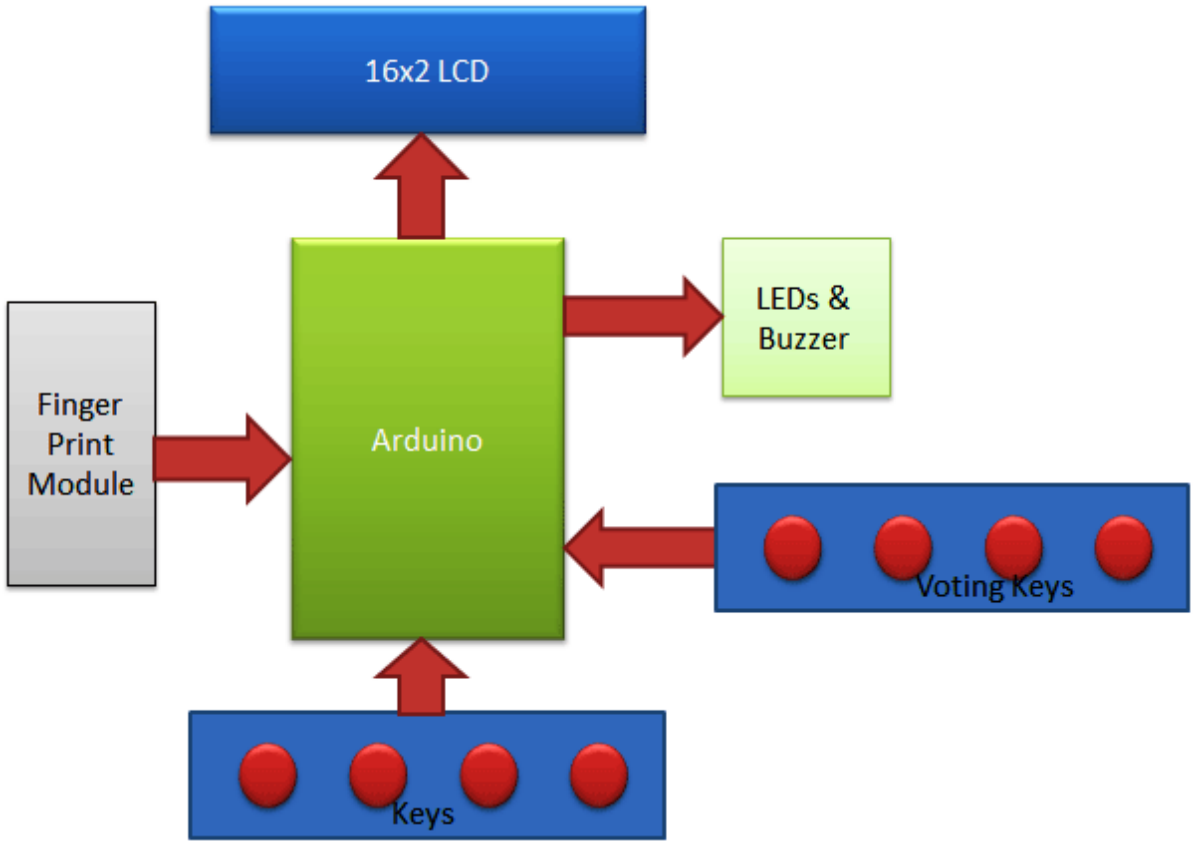



The circuit of this FingerPrint Based Voting Machine Project is very simple which contains Arduino for controlling whole the process of the project, push button for enrolling, deleting, selecting IDs and voting purpose, a buzzer for alert, LEDs for indication and 16x2 LCD for instruct Voter and showing the result as well. Yellow LED indicates that fingerprint module is ready to take an image of the finger and Green LED indicates that system is ready to take a vote or see results.

The push button is directly connected to pin A0(ENROL), A1(DEL), A2(UP), A3(DOWN) and A4(Match), D5(Can1), D4(Can2), D3(Can3), D2(Result) of Arduino with respect to ground. Yellow LED is connected at Digital pin D7 of Arduino with respect to ground through a 1k resistor and Green LED is connected to D6 of Arduino with the same method. Fingerprint module's Rx and Tx directly connected at Serial pin Tx and Rx of Arduino. 5v supply is used for powering finger print module taken from Arduino board. A buzzer is also connected at A5. A 16x2 LCD is configured in 4-bit mode and its RS, EN, D4, D5, D6, and D7 are directly connected at Digital pin D13, D12, D11, D10, D9, and D8 of Arduino.

**Working Explanation:**

Working of this **Biometric Voting System for Election** is a little bit complex for beginners. First of all, user needs to enroll finger or voters (in this code max limit of the voter is 25) with the help of push buttons/keys. To do this user need to press ENROLL key and then LCD asks for entering location/ID where finger will be a store. So now user needs to enter ID (Location) by using UP/DOWN keys. After selecting Location/ID user needs to press an OK key (DEL key). Now LCD will ask for placing finger over the finger print module. Now user needs to put his finger over finger print module. Then LCD will ask to remove the finger from finger print module and again ask for placing the finger. Now user needs to put his finger again over finger print module. Now finger print module takes an image and converts it into templates and stores it by selected ID in to the finger print module's memory. Now voter will be registered and he/she can vote. By same method all the voter can be registered into the system.




Now if the user wants to remove or delete any of stored ID then he/she need to press DEL key, after pressing DEL key, LCD will ask for select location means select ID that to be deleted.

Now user needs to select ID and press OK key (same DEL key). Now LCD will let you know that finger has been deleted successfully.

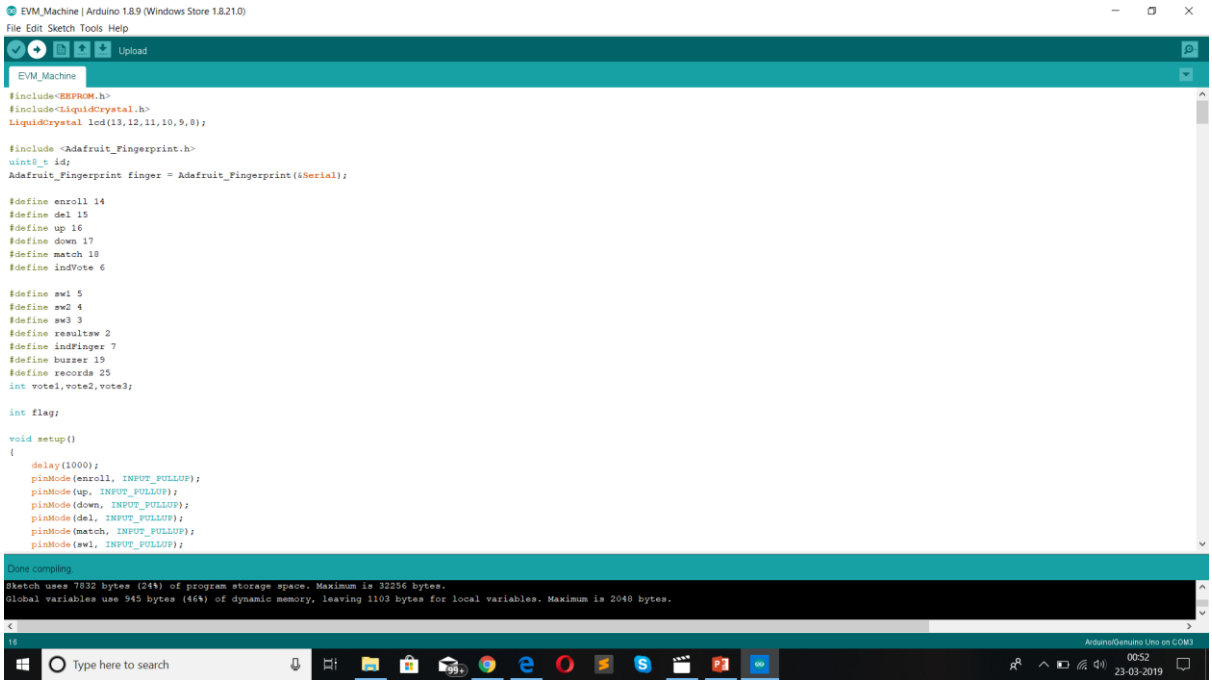
**Voting Process**:

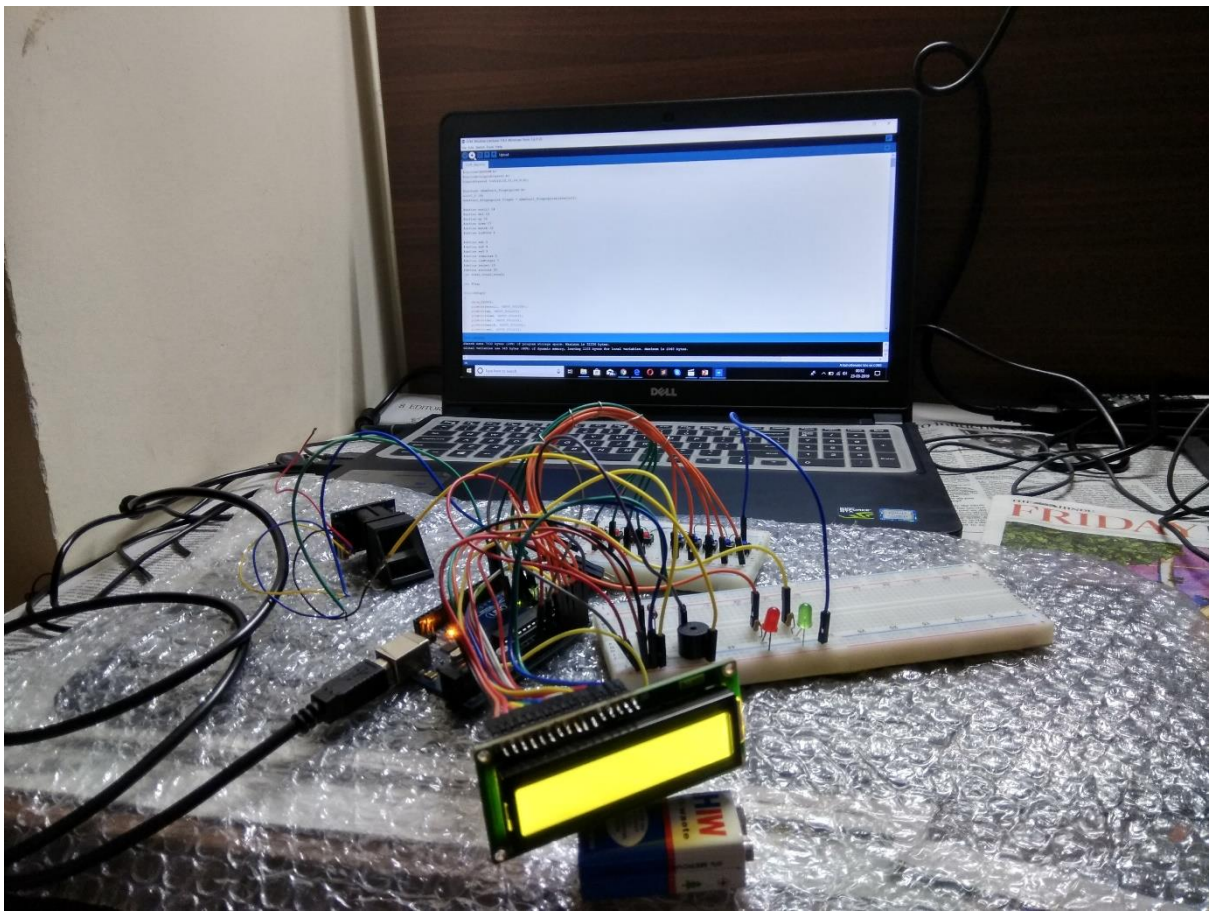
Now when user wants to vote then he/she needs to press match key and then buzzer will beep and LED will also glow and LCD will ask for place finger over fingerprint module. Now Arduino will give you three attempts to put your finger. After placing a finger over fingerprint module fingerprint module captures finger image find its IDs is present in the system. If finger ID detected then LCD will show authorized Voter. It means the user is authorized to vote. And then the system moves to next stage for voting. Now Green LED will glow it means now voter can vote for their candidates by pressing a relected key (from RED bread board in this demonstration). Now if the same voter wants to vote again then the system will show it *'Already Voted'*. Means same voter can't vote again and buzzer will beep for 5 seconds. If any Non-registered user wants to vote then finger print module will not detect its ID into the system and LCD will show '*No Fingerprint Found'*.


CAN1, CAN2, CAN3 here represents the Candidate 1, Candidate 2 and Candidate 3, who have stood for election

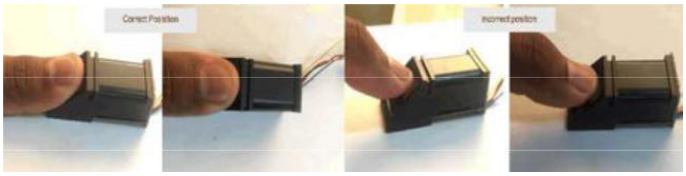
## Results and Discussion:

After writing down the code in the Arduino IDE, compile and upload it to the Arduino



First enrol the voter's finger and save the fingerprint by given id.



The above figure shows how to place finger on fingerprint module. The first two images were explained correct position and another two were wrong position of the fingerprint scanning.

In this time voter ask to user to get an id to save their fingerprint. After given id voter place their finger on fingerprint module to scan, during the enrolling voter place their finger in two times, in first time image take and convert, then second time check the fingerprint with first scan, if fingerprint matched save the fingerprint in given id. Otherwise "Fingerprint did not match" message displayed on LCD.

In this step voter scan their finger if fingerprint matched, LCD displayed a message "Did Not Match", then matched "Found Match", "Found Id" messages were displayed on LCD. In this step display the Id of saved fingerprint

In this step voters select a party to their preference from the party list, (Figure 15, 16) if select a party then cannot change to select another party

After select party, voters cast their three preferential votes to the candidates from the selected party (Figure 17).

If press party list button more the one time, it's not allowed to poll vote and cannot select more than 3 candidates from the candidate list

## Conclusion and Future Scope:

In total, this system overcomes most of the problems faced during the voting period by the paper ballot system. The efficiency of this system depends upon the web interface, its usability. This will surely ensure a safer voting method which is very much what is required for a healthy growth of a developing nation.

In this paper, the proposed Fingerprint based voting system which is better and faster than previous systems. The new system prevents access to illegal voters, provides ease of use,

transparency and maintains integrity of the voting process. The system also prevents multiple votes by the same person and checks eligibility of the voter. It also allows a person to vote from anywhere provided that the voter is within electoral limits.

Fingerprint based voting system has provided chance to avoid invalid votes, It reduce the polling time, Easy to carrying to polling center from the polling box, Reduce the staff of voting center, It provide easy and accurate counting without any troubles, Provisioning of voting preventive measures

**Future Scope:**

In big elections there are huge number of people want to cast their votes, in order to avoid the congestion at the voting point there is need to provide number of personal computers each one will be connected to the main computer/server in order to allow many people to perform voting at the same time and prevent congestion.

Therefore this application shouldbe built around server architecture. As an improvement multiple client machines should be interacting with the server simultaneously. Clients will interact with the system through an interactive GUI, while the server serves the clients request and does the processing in the backend.

This can be modify by interfacing it with a PC through a serial port in order to provide additional security

This fingerprint electronic voting system is considered as a PC based fingerprint voting system. For future work, it will be better to design a fingerprint voting machine works dependently without need for PC to perform the voting in order to decrease the project cost.

This project can be used as a voting machine to prevent rigging, during the elections in the polling booths.

- Fast track voting which could be used in small scale elections, like resident welfare association, “panchayat” level election and other society level elections, where results can be instantaneous.
- It could also be used to conduct opinion polls during annual shareholders meeting.
- It could also be used to conduct general assembly elections where number of candidates are less than or equal to eight in the current situation, on a small scale basis.

## References:

www.electronicsforu.com

www.electronicshub.org

www.elprocus.com

www.skyfilabs.com

www.nevonprojects.com

www.engineeringproject.com

## Codes in Appendix:

**Arduino Code:**

```
#include<EEPROM.h>
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

#include <Adafruit_Fingerprint.h>
uint8_t id;
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial);

#define enroll 14
#define del 15
#define up 16
#define down 17
#define match 18
#define indVote 6

#define sw1 5
#define sw2 4
#define sw3 3
#define resultsw 2
#define indFinger 7
#define buzzer 19
#define records 25
int vote1,vote2,vote3;

int flag;

void setup()
{
    delay(1000);
    pinMode(enroll, INPUT_PULLUP);
    pinMode(up, INPUT_PULLUP);
    pinMode(down, INPUT_PULLUP);
    pinMode(del, INPUT_PULLUP);
    pinMode(match, INPUT_PULLUP);
    pinMode(sw1, INPUT_PULLUP);
    pinMode(sw2, INPUT_PULLUP);
    pinMode(sw3, INPUT_PULLUP);
    pinMode(resultsw, INPUT_PULLUP);
    pinMode(buzzer, OUTPUT);
    pinMode(indVote, OUTPUT);
    pinMode(indFinger, OUTPUT);

lcd.begin(16,2);
  if(digitalRead(resultsw) ==0)
  {
```

```
      for(int i=0;i<records;i++)
         EEPROM.write(i+10,0xff);
      EEPROM.write(0,0);
      EEPROM.write(1,0);
      EEPROM.write(2,0);
      lcd.clear();
      lcd.print("System Reset");
      delay(1000);
 }

   lcd.clear();
   lcd.print("Voting Machine");
   lcd.setCursor(0,1);
   lcd.print("by Finger Print");
   delay(2000);
   lcd.clear();
   lcd.print("VIT-AP");
   lcd.setCursor(0,1);
   lcd.print("EC Expo");
   delay(2000);

 if(EEPROM.read(0) == 0xff)
  EEPROM.write(0,0);

    if(EEPROM.read(1) == 0xff)
  EEPROM.write(1,0);

    if(EEPROM.read(1) == 0xff)
  EEPROM.write(1,0);



   //finger.begin(57600);
   Serial.begin(57600);
   lcd.clear();
   lcd.print("Finding Module");
   lcd.setCursor(0,1);
   delay(1000);
   if (finger.verifyPassword())
   {
     //Serial.println("Found fingerprint sensor!");
     lcd.clear();
     lcd.print("Found Module ");
     delay(1000);
   }
   else
   {
   //Serial.println("Did not find fingerprint sensor :(");
   lcd.clear();
   lcd.print("module not Found");
   lcd.setCursor(0,1);
   lcd.print("Check Connections");
   while (1);
   }
```

```
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Cn1");
  lcd.setCursor(4,0);
  lcd.print("Cn2");
  lcd.setCursor(8,0);
  lcd.print("Cn3");
  lcd.setCursor(12,0);
  lcd.print("Cn4");

  lcd.setCursor(0,1);
  vote1=EEPROM.read(0);
  lcd.print(vote1);
  lcd.setCursor(6,1);
  vote2=EEPROM.read(1);
  lcd.print(vote2);
  lcd.setCursor(12,1);
  vote3=EEPROM.read(2);
  lcd.print(vote3);
  delay(2000);
}

void loop()
{
 lcd.setCursor(0,0);
 lcd.print("Press Match Key ");
 lcd.setCursor(0,1);
 lcd.print("to start system");

 digitalWrite(indVote, LOW);
 digitalWrite(indFinger, LOW);
 if(digitalRead(match)==0)
 {
  digitalWrite(buzzer, HIGH);
  delay(200);
  digitalWrite(buzzer, LOW);
  digitalWrite(indFinger, HIGH);
  for(int i=0;i<3;i++)
  {
     lcd.clear();
    lcd.print("Place Finger");
    delay(2000);
    int result=getFingerprintIDez();
    if(result>=0)
    {
      flag=0;
        for(int i=0;i<records;i++)
        {
          if(result == EEPROM.read(i+10))
          {
             lcd.clear();
              lcd.print("Authorised Voter");
              lcd.setCursor(0,1);
```

```
              lcd.print("Please Wait....");
              delay(1000);
              Vote();
              EEPROM.write(i+10, 0xff);
              flag=1;
              return;
            }
          }

          if(flag == 0)
          {
          lcd.clear();
          lcd.print("Already Voted");
          //lcd.setCursor(0,1);
          //lcd.print("")
          digitalWrite(buzzer, HIGH);
          delay(5000);
          digitalWrite(buzzer, LOW);
          return;
          }
       }
     }
     lcd.clear();
  }
  checkKeys();
  delay(1000);
}

void checkKeys()
{
   if(digitalRead(enroll) == 0)
   {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
    while(digitalRead(enroll) == 0);
    Enroll();
   }

   else if(digitalRead(del) == 0)
   {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
    delet();
   }
}

void Enroll()
{
   int count=0;
   lcd.clear();
   lcd.print("Enter Finger ID:");
```

```
   while(1)
   {
    lcd.setCursor(0,1);
     lcd.print(count);
     if(digitalRead(up) == 0)
     {
       count++;
       if(count>25)
       count=0;
       delay(500);
     }

     else if(digitalRead(down) == 0)
     {
       count--;
       if(count<0)
       count=25;
       delay(500);
     }
     else if(digitalRead(del) == 0)
     {
          id=count;
          getFingerprintEnroll();
          for(int i=0;i<records;i++)
          {
            if(EEPROM.read(i+10) == 0xff)
            {
              EEPROM.write(i+10, id);
              break;
            }
          }
          return;
     }

       else if(digitalRead(enroll) == 0)
     {
          return;
     }
 }
}

void delet()
{
   int count=0;
   lcd.clear();
   lcd.print("Enter Finger ID");

   while(1)
   {
    lcd.setCursor(0,1);
     lcd.print(count);
     if(digitalRead(up) == 0)
     {
       count++;
```

```
       if(count>25)
       count=0;
       delay(500);
     }

     else if(digitalRead(down) == 0)
     {
       count--;
       if(count<0)
       count=25;
       delay(500);
     }
     else if(digitalRead(del) == 0)
     {
          id=count;
          deleteFingerprint(id);
          for(int i=0;i<records;i++)
          {
            if(EEPROM.read(i+10) == id)
            {
              EEPROM.write(i+10, 0xff);
              break;
            }
          }
          return;
     }
       else if(digitalRead(enroll) == 0)
     {
          return;
     }
 }
}

uint8_t getFingerprintEnroll()
{
  int p = -1;
  lcd.clear();
  lcd.print("finger ID:");
  lcd.print(id);
  lcd.setCursor(0,1);
  lcd.print("Place Finger");
  delay(2000);
  while (p != FINGERPRINT_OK)
  {
    p = finger.getImage();
    switch (p)
    {
    case FINGERPRINT_OK:
      //Serial.println("Image taken");
      lcd.clear();
      lcd.print("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      //Serial.println("No Finger");
```

```
      lcd.clear();
      lcd.print("No Finger");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      //Serial.println("Communication error");
      lcd.clear();
      lcd.print("Comm Error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      //Serial.println("Imaging error");
      lcd.clear();
      lcd.print("Imaging Error");
      break;
    default:
      //Serial.println("Unknown error");
       lcd.clear();
      lcd.print("Unknown Error");
      break;
    }
  }

  // OK success!

  p = finger.image2Tz(1);
  switch (p) {
    case FINGERPRINT_OK:
      //Serial.println("Image converted");
      lcd.clear();
      lcd.print("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
      //Serial.println("Image too messy");
       lcd.clear();
       lcd.print("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      //Serial.println("Communication error");
            lcd.clear();
      lcd.print("Comm Error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      //Serial.println("Could not find fingerprint features");
            lcd.clear();
      lcd.print("Feature Not Found");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      //Serial.println("Could not find fingerprint features");
                  lcd.clear();
      lcd.print("Feature Not Found");
      return p;
    default:
      //Serial.println("Unknown error");
                  lcd.clear();
      lcd.print("Unknown Error");
```

```
    return p;
  }

  //Serial.println("Remove finger");
  lcd.clear();
  lcd.print("Remove Finger");
  delay(2000);
  p = 0;
  while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
  }
  //Serial.print("ID "); //Serial.println(id);
  p = -1;
  //Serial.println("Place same finger again");
   lcd.clear();
      lcd.print("Place Finger");
      lcd.setCursor(0,1);
      lcd.print("   Again");
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
      //Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      //Serial.print(".");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      //Serial.println("Communication error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      //Serial.println("Imaging error");
      break;
    default:
      //Serial.println("Unknown error");
      return;
    }
  }

  // OK success!

  p = finger.image2Tz(2);
  switch (p) {
    case FINGERPRINT_OK:
      //Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
      //Serial.println("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      //Serial.println("Communication error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      //Serial.println("Could not find fingerprint features");
```

```
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      //Serial.println("Could not find fingerprint features");
      return p;
    default:
      //Serial.println("Unknown error");
      return p;
  }

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

  p = finger.createModel();
  if (p == FINGERPRINT_OK) {
    //Serial.println("Prints matched!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    //Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    //Serial.println("Fingerprints did not match");
    return p;
  } else {
    //Serial.println("Unknown error");
    return p;
  }

  //Serial.print("ID "); //Serial.println(id);
  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
    //Serial.println("Stored!");
    lcd.clear();
    lcd.print("Stored!");
    delay(2000);
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    //Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    //Serial.println("Could not store in that location");
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    //Serial.println("Error writing to flash");
    return p;
  }
  else {
    //Serial.println("Unknown error");
    return p;
  }
}

int getFingerprintIDez()
{
  uint8_t p = finger.getImage();

  if (p != FINGERPRINT_OK)
```

```
  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)
  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)
  {
   lcd.clear();
   lcd.print("Finger Not Found");
   lcd.setCursor(0,1);
   lcd.print("Try Later");
   delay(2000);
  return -1;
  }
  // found a match!
  //Serial.print("Found ID #");
  //Serial.print(finger.fingerID);
  return finger.fingerID;
}

uint8_t deleteFingerprint(uint8_t id)
{
  uint8_t p = -1;
  lcd.clear();
  lcd.print("Please wait");
  p = finger.deleteModel(id);
  if (p == FINGERPRINT_OK)
  {
    //Serial.println("Deleted!");
    lcd.clear();
    lcd.print("Figer Deleted");
    lcd.setCursor(0,1);
    lcd.print("Successfully");
    delay(1000);
  }

  else
  {
    //Serial.print("Something Wrong");
    lcd.clear();
    lcd.print("Something Wrong");
    lcd.setCursor(0,1);
    lcd.print("Try Again Later");
    delay(2000);
    return p;
  }
}

void Vote()
{
  lcd.clear();
  lcd.print("Please Place");
```

```
  lcd.setCursor(0,1);
  lcd.print("Your Vote");
  digitalWrite(indVote, HIGH);
  digitalWrite(indFinger, LOW);
  digitalWrite(buzzer, HIGH);
  delay(500);
  digitalWrite(buzzer, LOW);
  delay(1000);
  while(1)
  {
        if(digitalRead(sw1)==0)
        {
          vote1++;
          voteSubmit(1);
          EEPROM.write(0, vote1);
          while(digitalRead(sw1)==0);
          return;
        }
        if(digitalRead(sw2)==0)
        {
          vote2++;
            voteSubmit(2);
            EEPROM.write(1, vote2);
          while(digitalRead(sw2)==0);
          return;
        }
        if(digitalRead(sw3)==0)
        {
          vote3++;
             voteSubmit(3);
             EEPROM.write(2, vote3);
          while(digitalRead(sw3)==0);
          return;
        }

         if(digitalRead(resultsw)==0)
         {
              lcd.clear();
              lcd.setCursor(0,0);
              lcd.print("Can1");
              lcd.setCursor(6,0);
              lcd.print("Can2");
              lcd.setCursor(12,0);
              lcd.print("Can3");
              for(int i=0;i<3;i++)
              {
                lcd.setCursor(i*6,1);
                lcd.print(EEPROM.read(i));
              }
              delay(2000);
           int vote=vote1+vote2+vote3;
           if(vote)
           {
            if((vote1 > vote2 && vote1 > vote3))
```

```
            {
             lcd.clear();
             lcd.print("Can1 Wins");
             delay(2000);
             lcd.clear();
            }
            else if(vote2 > vote1 && vote2 > vote3)
            {
             lcd.clear();
             lcd.print("Can2 Wins");
             delay(2000);
             lcd.clear();
            }
            else if((vote3 > vote1 && vote3 > vote2))
            {
             lcd.clear();
             lcd.print("Can3 Wins");
             delay(2000);
             lcd.clear();
            }

           else
           {
             lcd.clear();
             lcd.print("   Tie Up Or   ");
             lcd.setCursor(0,1);
             lcd.print("   No Result   ");
             delay(1000);
             lcd.clear();
           }

          }
           else
           {
             lcd.clear();
             lcd.print("No Voting....");
             delay(1000);
             lcd.clear();
           }
           vote1=0;vote2=0;vote3=0;vote=0;
           lcd.clear();
           return;
         }
  }
   digitalWrite(indVote, LOW);
}

void voteSubmit(int cn)
{
  lcd.clear();
  if(cn == 1)
      lcd.print("Can1");
  else if(cn == 2)
      lcd.print("Can2");
```

```
  else if(cn == 3)
      lcd.print("Can3");
  lcd.setCursor(0,1);
  lcd.print("Vote Submitted");
  digitalWrite(buzzer , HIGH);
  delay(1000);
  digitalWrite(buzzer, LOW);
  digitalWrite(indVote, LOW);
  return;
}
```