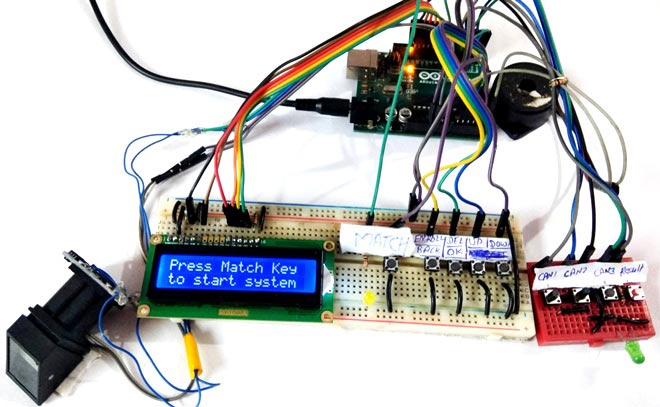
**Fingerprint Based Biometric Voting Machine using Arduino**



In the realm of modern elections, Electronic Voting Machines have become ubiquitous, streamlining the voting process by digitizing ballot registration. However, the paramount concern remains security, particularly in preventing instances of double voting and eliminating the potential for fraudulent practices. In response to this challenge, we present the FingerPrint Based Biometric Voting Machine using Arduino. This innovative system leverages biometric authentication through fingerprint recognition, providing a secure and tamper-resistant method to authorize voters. By linking each vote to a unique fingerprint, the technology ensures the integrity of the electoral process and effectively combats the threat of unauthorized or duplicate voting. Join us in exploring this cutting-edge application of Arduino technology to enhance the security and reliability of the voting experience.  
  
  
  
Required Components:

1. Arduino Uno
2. Finger Print Sensor Module
3. Push Buttons
4. LEDs -2
5. 1K Resistor -3
6. 2.2K resistor
7. Power
8. Connecting wires
9. Buzzer
10. 16x2 LCD
11. Bread-Board / PCB

Finger Print Sensor Module in Voting Machine:

The Finger Print Sensor Module, or Finger Print Scanner, serves as a crucial component in this system, tasked with capturing the unique fingerprint image. Following capture, the module undertakes the intricate process of transforming the fingerprint image into a corresponding template. This template is then meticulously preserved in the module's memory under a designated ID, a task overseen and directed by the Arduino. Every step of this intricate operation, from image acquisition to template conversion and storage location assignment, is orchestrated by the Arduino. In essence, the Arduino assumes command, ensuring the seamless execution of each facet of the fingerprint recognition and storage process.



In this FingerPrint Voting Machine Circuit, the authentication process is facilitated by the Finger Print Sensor Module, which captures and processes the user's fingerprint input. The circuit incorporates 5 push buttons, each serving a distinct purpose: Match, Enroll/Back, Delete/OK, UP, and Down. Notably, the Enroll and Delete keys possess dual functionality. The Enroll key functions both as the trigger for enrolling new fingerprints and as a back button. If a user decides not to proceed after pressing the Enroll key during the ID/location selection phase, pressing it again acts as a back button, allowing the user to navigate to the previous step.

Similarly, the DEL/OK key also has a dual role. After selecting the ID or location using the UP and DOWN keys, pressing DEL/OK serves as confirmation (OK) to proceed with the chosen ID or location. Meanwhile, the Match key is pivotal for voters intending to cast their votes. Before voting, the user must authenticate themselves by placing their finger on the Finger Print Sensor. Successful authentication grants the user the privilege to vote.

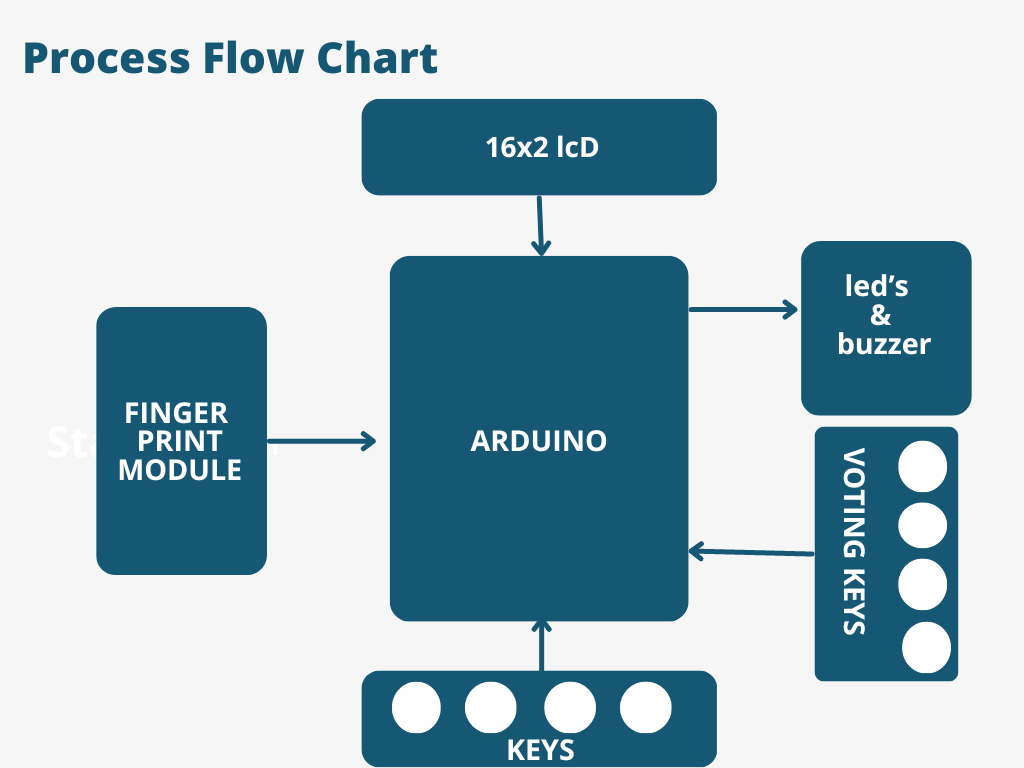
This design intricately weaves together these functions, ensuring a user-friendly and efficient experience in the fingerprint-based voting process. The Enroll and Delete keys offer flexibility, serving both primary functions seamlessly, while the Match key acts as the gateway to the voting process, ensuring the security and legitimacy of each vote cast.

### Working :

The operational flow of the Biometric Voting System may initially seem intricate for beginners. To initiate the system, a user is required to enroll voters' fingerprints, with a set maximum limit of 25 voters in this particular code. The process begins with the user pressing the 'ENROLL' key, prompting the LCD to request the entry of a unique location or ID for storing the fingerprint data. Using the 'UP/DOWN' keys, the user selects the desired location and confirms the choice by pressing the 'OK' (DEL) key.

Subsequently, the LCD instructs the user to place their finger over the fingerprint module. Following this prompt, the user complies, allowing the system to capture an image of the fingerprint. The system then prompts the user to remove their finger from the module, followed by another request to place the finger once more. This iterative process ensures the system captures multiple perspectives of the fingerprint, enhancing accuracy.

The fingerprint module processes the collected images, converting them into templates that are then stored in the module's memory under the selected ID. With this, the voter is successfully registered, gaining the authorization to cast their vote. This systematic approach ensures the enrollment of all voters, securing the integrity of the Biometric Voting System.



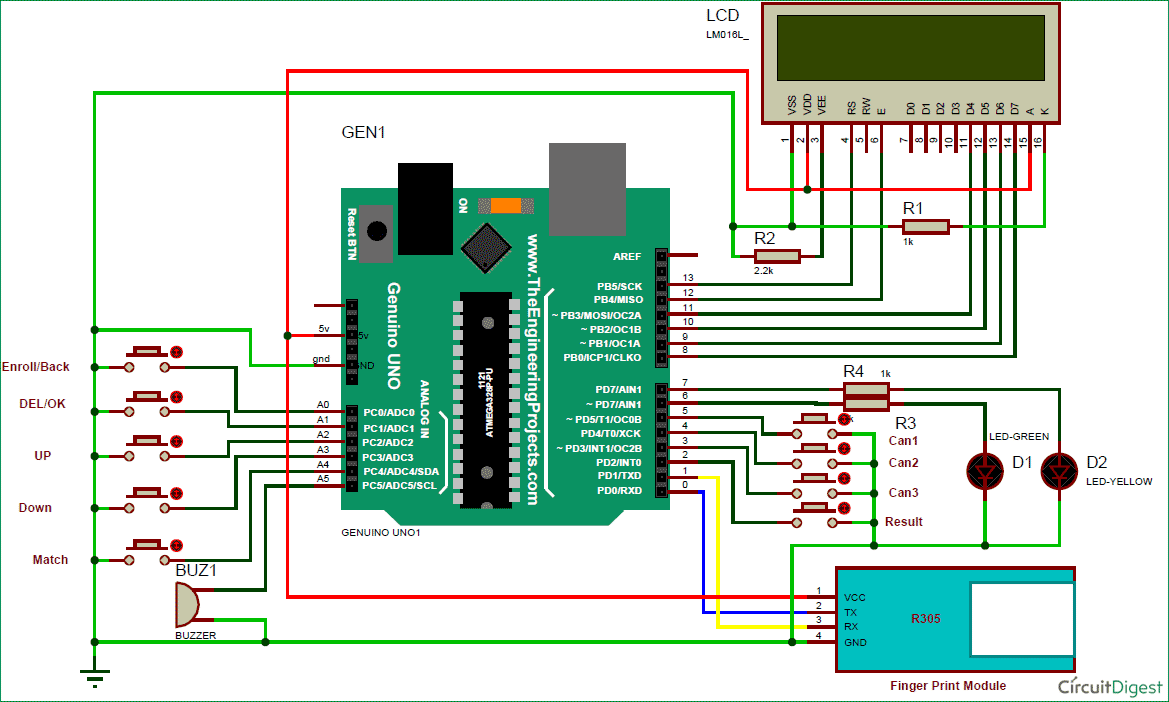
Now if the user wants to remove or delete any of stored ID then he/she need to press DEL key, after pressing DEL key, LCD will ask for select location means select ID that to be deleted. Now user needs to select ID and press OK key (same DEL key). Now LCD will let you know that finger has been deleted successfully.  
  
  
**Voting Process:**

When a user desires to cast their vote, the initiation begins with the pressing of the Match key. Subsequently, a buzzer emits a distinctive beep, accompanied by the illumination of an LED, signaling the start of the authentication process. Simultaneously, the LCD prompts the user to position their finger over the fingerprint module. The Arduino extends three attempts for the user to place their finger successfully.

Upon fingerprint placement, the module captures the fingerprint image, initiating the search for a matching ID within the system. If the system identifies a matching fingerprint ID, the LCD promptly displays 'Authorized Voter,' signifying the user's eligibility to cast their vote. At this juncture, the system transitions to the voting stage, indicated by the activation of a Green LED. The voter can now make their candidate selections by pressing the corresponding keys on the RED breadboard.

In the event that a previously authorized voter attempts to cast another vote, the system intelligently recognizes this and displays 'Already Voted' on the LCD. Simultaneously, a buzzer emits a 5-second beep, emphasizing the system's prohibition against multiple votes from the same individual. Additionally, if a non-registered user attempts to vote, the fingerprint module fails to detect their ID, prompting the LCD to display 'No Fingerprint Found.' This stringent authentication process ensures the integrity of the voting system by preventing unauthorized or duplicate votes.

**CIRCUIT DIAGRAM:**



Referenced by: CircuitDigest

**CODE :**

#include <EEPROM.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

#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 2

#define sw3 3

#define resultsw 4

#define indFinger 7

#define buzzer 6

#define records 25

int votes[3] = {0};

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) {

resetSystem();

}

displayWelcomeScreen();

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

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

if (EEPROM.read(2) == 0xff) EEPROM.write(2, 0);

initializeFingerprintModule();

displayCandidateVotes();

}

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) {

initializeVotingProcess();

}

checkKeys();

delay(1000);

}

void checkKeys() {

if (digitalRead(enroll) == 0) {

lcd.clear();

lcd.print("Please Wait");

delay(1000);

while (digitalRead(enroll) == 0)

;

enrollVoter();

} else if (digitalRead(del) == 0) {

lcd.clear();

lcd.print("Please Wait");

delay(1000);

deleteVoter();

}

}

void enrollVoter() {

int count = 0;

lcd.clear();

lcd.print("Enter Finger ID:");

while (1) {

lcd.setCursor(0, 1);

lcd.print(count);

if (digitalRead(up) == 0) {

count = (count + 1) % 26;

delay(500);

} else if (digitalRead(down) == 0) {

count = (count - 1 + 26) % 26;

delay(500);

} else if (digitalRead(del) == 0) {

id = count;

getFingerprintEnroll();

for (int i = 0; i < records; i++) {

if (id == EEPROM.read(i + 10)) {

lcd.clear();

lcd.print("Authorized Voter");

lcd.setCursor(0, 1);

lcd.print("Please Wait....");

delay(1000);

vote();

EEPROM.write(i + 10, 0xff);

return;

}

}

lcd.clear();

lcd.print("Already Voted");

buzzerBeep();

delay(5000);

return;

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

return;

}

}

}

void deleteVoter() {

int count = 0;

lcd.clear();

lcd.print("Enter Finger ID");

while (1) {

lcd.setCursor(0, 1);

lcd.print(count);

if (digitalRead(up) == 0) {

count = (count + 1) % 26;

delay(500);

} else if (digitalRead(down) == 0) {

count = (count - 1 + 26) % 26;

delay(500);

} else if (digitalRead(del) == 0) {

id = count;

deleteFingerprint(id);

for (int i = 0; i < records; i++) {

if (id == EEPROM.read(i + 10)) {

EEPROM.write(i + 10, 0xff);

break;

}

}

return;

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

return;

}

}

}

void initializeVotingProcess() {

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("Authorized 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");

buzzerBeep();

delay(5000);

return;

}

}

}

lcd.clear();