Biometrically Secured Electronic Voting Machine

May 5, 2018 **Embedded System Design** **CSE 671**

Sachin S. Vaze MSEE S.U.I.D. - 468358107

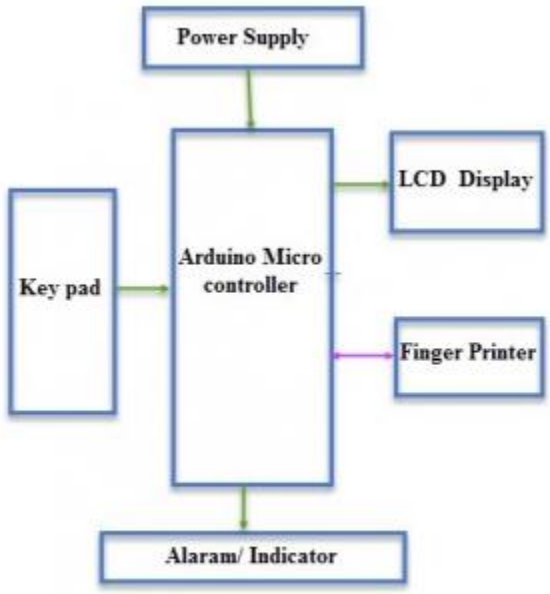
Abstract

The e-voting in general is known as electronic voting. It is a voting process developed by using the electronic resources for simplification and it can perform effortlessly. The ranges of internet services are included in e-voting and it depends on the situation. The amount of automation may vary from simple task to a solution that can have voter registration and authentication, voter input, vote date services, tabulation and administration of elections. Admirable e-voting system should do the most number of tasks and fulfill the set of standard establishments through regular bodies.

Biometric Voting System has been implemented using the Arduino and Fingerprint sensor. In this system, a voter can poll his/her vote very easily. Also in cloud database server, all voters' information gets stored. The voter should fill a registration form that has a voter ID (fingerprint ID) and the corresponding name. If anything is wrong, it perfectly prevents false voting. So, it is hack proof, both hardware & software wise. Also, all the information will be updated at the cloud server at real time. This system is user-friendly & decreases the time of voting process also. It is well secured as fingerprint is an important identity of the user. It has simple architecture. It provides easy and accurate counting without any troubles. It automatically generates the result. It is portable & cost effective.

Introduction

Pictorial representation of a Fingerprint based Electronic Voting system:



Block Diagram of Biometric Voting Machine

How Arduino is useful?

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or our

TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a huge variety of Arduino-based projects.

The initial approach was to develop the project using Beaglebone Black as it provides an excellent client-server communication and a good learning experience with Linux based embedded systems which would have also helped to establish cloud storage demonstration in a good way. I was able to interface LCD display and libraries required for fingerprint sensor using Beaglebone black. But unfortunately the Beaglebone hardware was crashed and the project deadline could not have been met while troubleshooting it. So, finally Arduino Nano was the best low cost choice for me.

Basic outline of project:

The main objective of this project is to integrate a low cost EVM system which can perform very nicely and can provide excellent security and data storage features. The goal of the project is developing a hack-proof EVM; both hardware & software wise. The key features are:

1) Fingerprint based voter registration

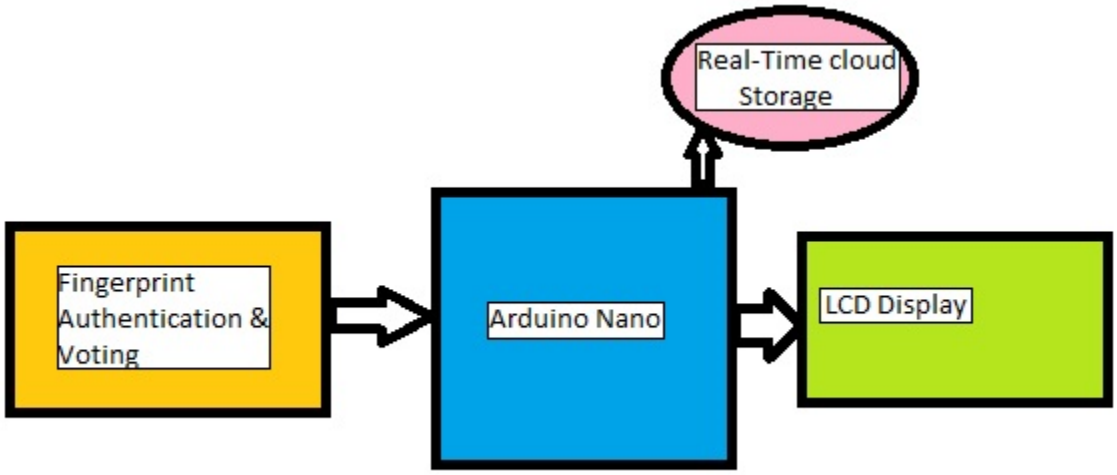
2) Fingerprint verification

3) User-friendly Electronic voting

4) Excellent security to avoid false voting

5) Automatic data processing, ballot counting and cloud storage

In this project a simple user-friendly EVM was constructed. With time constraints and hardware complexity in mind, the output was implemented through LCD display.



Background

The fingerprint based voting project demands the user to submit Fingerprint at the polling booth. The orthodox voting systems were very much time-consuming and hassle prone. But this embedded electronic approach is novel in terms of overall performance and ease of conduction. The voting is very sensitive thing and has to be handled with very much precision. This Biometric EVM is a good to build option for the government bodies with the best possible outfit to carry out elections.

Out of various available sensors I chose fingerprint sensor because it is cheap and serves the purpose extremely well. The thumb fingerprint of every person is considered as the most unique pattern. Moreover this system is very much user-friendly, as the users can ensure that their 'single' and valuable vote is casted successfully. Also with the specific hardware and software structure, the system is hack-proof. The cloud data storage is also a very strong aspect of this project as the large amount of data is easily accessible, updatable from anywhere, anytime and that too with the excellent security features. It is not a difficult task for the government bodies or election authorities to build their own cloud server system. For the school level demo, I was going to demonstrate the cloud feature using currently available cloud platforms. The internet connectivity is now-a-days a very trivial thing. Even our smart-phones also continuously need internet connectivity otherwise they are not really 'smart'.

Brief user interface and working of my project –

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 user to show his/her Fingerprint as an identity proof. The system reads the data from the voter's Fingerprint and verifies the data which is already stored data in the database. Before this, obviously the voters need to register their fingerprints at the polling center. If the given details match with the database data, the system allows the person to cast their vote by showing "Authorized ID: XX" message on LCD display. If the given Fingerprint data does not match with the stored data, the system immediately activates the display as "Finger Not Found". He/she maybe given another chance to try again since sometimes there could be some problem with the system to verify the provided data. But, if still the voter's fingerprint is not verified again & again then his/her claimed name and other credentials have to be verified with our original database. So, the voter with no prior fingerprint registration cannot vote, the security authorities take the further action.

In case a registered voter gets verified but somehow could not understand the procedure and is unable to/forgot to cast the vote, he/she will be asked to follow the procedure again. The system takes care of 'false voting' scenarios very nicely. If the same voter comes back to cast the vote again, after scanning his/her fingerprint, a clear warning message "Already Voted" is displayed on the LCD screen. Also, each voter gets very limited time of only 5 seconds when asked to cast the vote and the push buttons used

for voting process are also implemented in such a way that one cannot simply press multiple buttons. The button has to be kept pressed for 5 seconds for successful submission of the vote. One has to keep a button pressed till the message “Vote Submitted. Thank you!” is displayed on the LCD display.

After the voting process is done the authority has been provided with a simple hardware button which when pressed gives the detailed display of voting took place at the center. It also quickly shows the current result. For large data, cloud storage is an excellent option which I’ve tried to implement further by using Wi-Fi module integration and ‘Arduino cloudino’ or ‘Google Firebase cloud platform’ interlinking with the current system which I could not really got working in the end. But as discussed earlier, it could have been easier to set it up using Beaglebone Black board. The best part about these cloud management systems is that the data gets updated at real-time.

It is one of my future prospects to achieve the cloud demonstration with my current biometric EVM.

So, the proposed Biometric EVM demonstrates a very secure, quick responsive, user-friendly, cost effective, self-processing and portable embedded system.

Methodology/Technical Approach

The project uses the Fingerprint scanner GT511C1R and Arduino Nano microcontroller board based on ATmega 328P processor. The Arduino based code was written in Arduino C language supported by Arduino IDE. A 16*2 LCD display model DEV-13293 from SparkFun Electronics was chosen. The system first needs to register the fingerprints of all the voters. The scanner that I used can store 20 fingerprints. The more costly model we will choose, the more number of fingerprints (unique voter IDs) can be stored. To do this I have used the Arduino library 'Fingerprint_Scanner-TTL-master' freely available on GitHub platform. [https://github.com/sparkfun/Fingerprint_Scanner-TTL]. Then I used the software serial library of Arduino to build serial communication between Arduino Nano and GT511C1R. After that, simple SDK Demo software specifically designed to work with the fingerprint scanner GT511C1R is used namely "GT-511C1R_SDK_20140312". This is used to test enrolling, verification, fingerprint image retrieving etc. very efficiently. One has to put his/her finger 3 times to successfully enroll and store the ID inside fingerprint module's memory. Along with that, it's possible to store voter ID, name and related documents on a cloud server as stated before.

After all the voters successfully calibrate their fingerprint IDs, we can move ahead with our project.

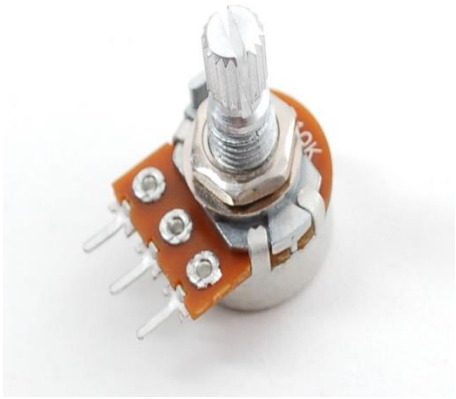
To dig into more into the technical details we shall look how each and every part coordinates with the system environment.

Arduino Nano –

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. It is super cheap and efficient option to build a good prototype of embedded electronic systems.

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328P provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digital pins. The ATmega328P also support I2C (TWI) and SPI communication.

16*2 LCD Display –



The LCD and Keypad Shield gives you a handy 16-character by 2-line display, 5 buttons and a controllable backlight, plug it straight in on top of your Arduino board or other project shields. The display is set behind the shield for a low profile fitment and nice look and we've included panel mounting screw holes in the corners.

It's great when you want to build a stand-alone project with its own user interface that doesn't require a computer attached to send commands to your Arduino.

It works perfectly in 4-bit mode with the "LiquidCrystal" library included with the Arduino IDE, allowing you to control the LCD with a total of just 6 digital I/O lines. We've deliberately picked D4-D9 so that it doesn't interfere with pins required by other popular products such as the Ethernet Shield and EtherTen, so you can stack this on top of other shields to give you a local display.

There are 5 buttons, and when press any combination of these buttons, each will give a unique value. That means that it is able to tell 32 different possibilities.

The LCD backlight can be controlled for on/off by Arduino program, and the 10K potentiometer is used to adjust the contrast of the display.

Fingerprint Sensor GT511C1R –

The fingerprint scanner requires a serial UART connection and power. There are a few options to connect to the sensor depending on what UART device. The easiest would be to use an FTDI but any microcontroller that has a UART can be also used. Since Arduino Nano comes with an inbuilt FTDI chip, it's very easy to interface this device.

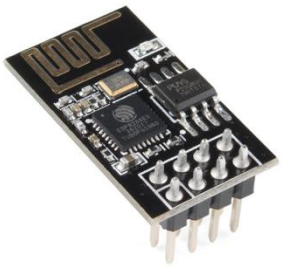
Push Buttons –

All these hardware components were successfully interfaced with Arduino Nano and the code was written using Arduino IDE with Arduino C.

Jumper cables and connecting wires –

In order to build the circuit connections, all types of cables/wires like 'male-to-male', 'male-to-female', 'female-to-female' etc. were used along with Arduino Nano USB to mini USB jack.

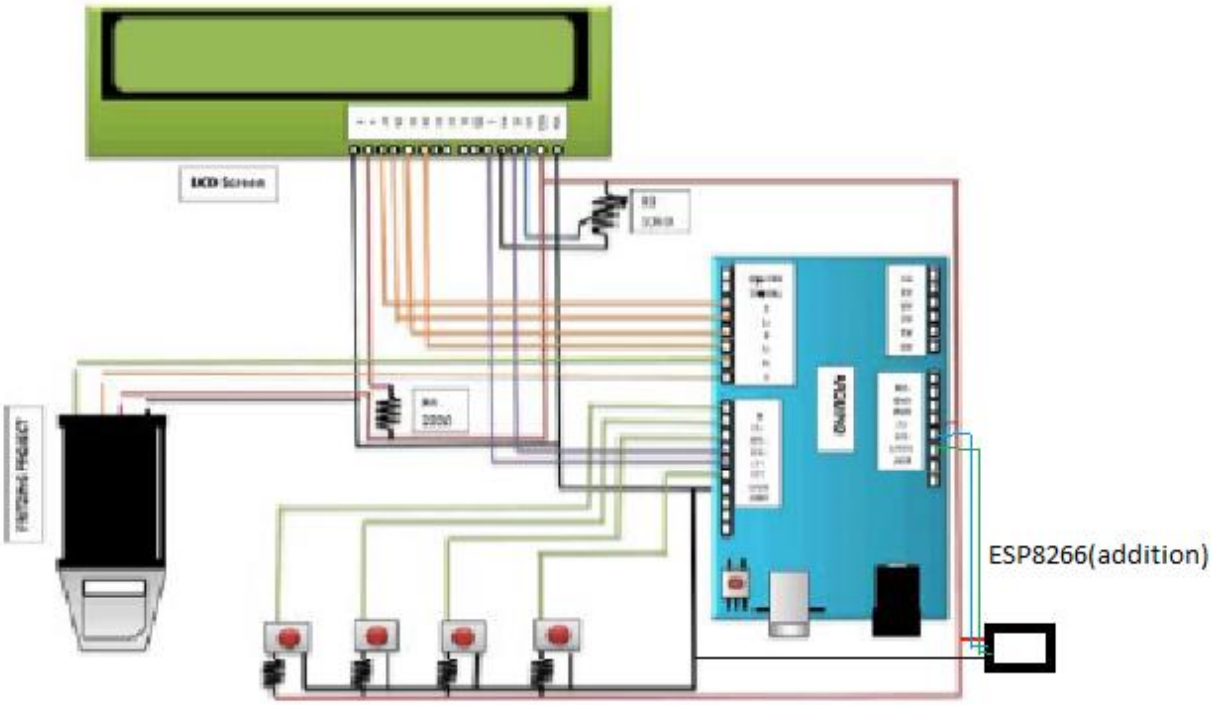
ESP8266 Wi-Fi Module –



ESP8266 is a complete and self-contained Wi-Fi network solution that can carry software applications, or through another application processor uninstall all Wi-Fi networking capabilities. When the device is mounted and as the only application of the application processor, the flash memory can be started directly from an external Move. Built-in cache memory will help improve system performance and reduce memory requirements. Another situation is when wireless Internet access assumes the task of Wi-Fi adapter, you can add it to any microcontroller-based design, and the connection is simple, just by SPI / SDIO interface or central processor AHB bridge interface. Processing and storage capacity on ESP8266 powerful piece, it can be integrated via GPIO ports sensors and other applications specific equipment to achieve the lowest early in the development and operation of at least occupy system resources. The ESP8266 highly integrated chip, including antenna switch, power management converter, so with minimal external circuitry, and includes front-end module, including the entire solution designed to minimize the space occupied by PCB. The system is equipped with ESP8266 manifested leading features are: energy saving VoIP quickly switch between the sleep / wake patterns, with low-power operation adaptive radio bias, front-end signal processing functions, troubleshooting and radio systems coexist characteristics eliminate cellular / Bluetooth / DDR / LVDS / LCD interference.

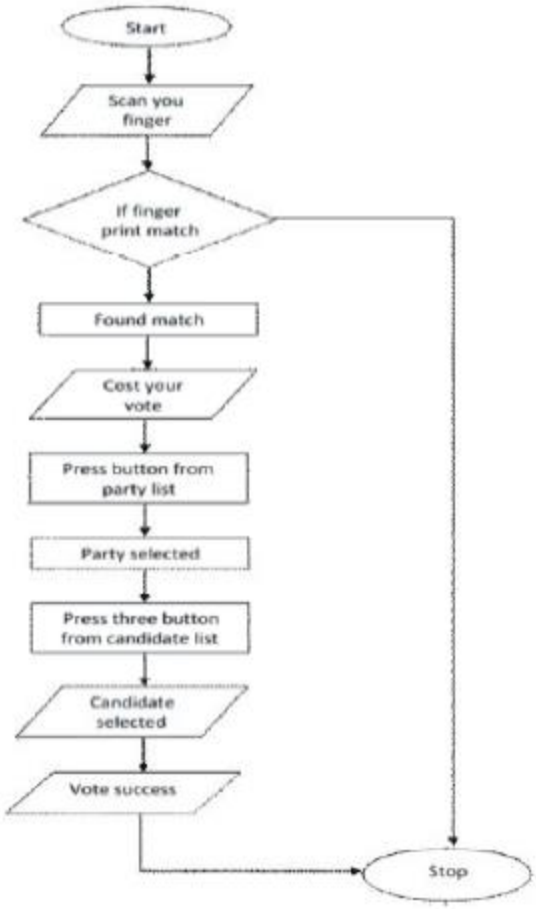
As stated earlier I tried to interface this chip with the Arduino Nano to achieve the cloud storage demo. But unfortunately it was not successful. Also Linux Distribution and Cloud9 IDE provided by Beaglebone black were helpful to construct the system and also the cloud platform interlinking may become easier.

Schematic Circuit Interfacing Diagram –



Along with the above things, the software used is Arduino IDE. Codes were written using Arduino C. My code has been written in Appendix part of this report.

Design rationale



Flow diagram of Biometric EVM system

The Fingerprint Voting System as the basis –‘ One Person – One Vote’, it stands to reason that must verify that a voter is who they claim to be and that they have not previously voted in this election at another site (to eliminate double voting) . The main purpose of fingerprint voting system is to ‘Preventing Fraudulent Voting’. This system has basically 5 types of modules as below –

- Fingerprint Enrollment
- Fingerprint Verification
- Cast the votes
- Alert for wrong voting
- Generate final report and cloud storage

People can trust the result because it allows for a process that is so auditable, transparent and secure. It also helps reduce human error. Fingerprint authentication, electronic counting plus the cloud storage together empowers this embedded design to the best possible optimization.

Challenges and alternatives:

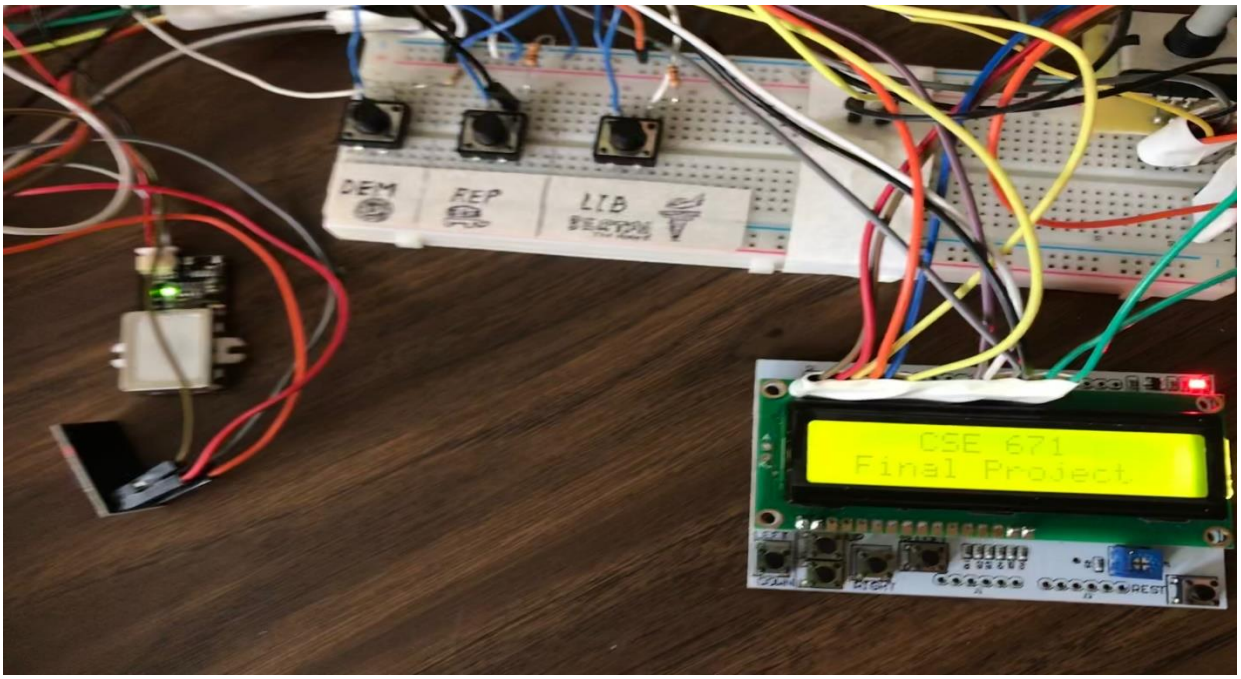
Some challenges faced by me are listed below:

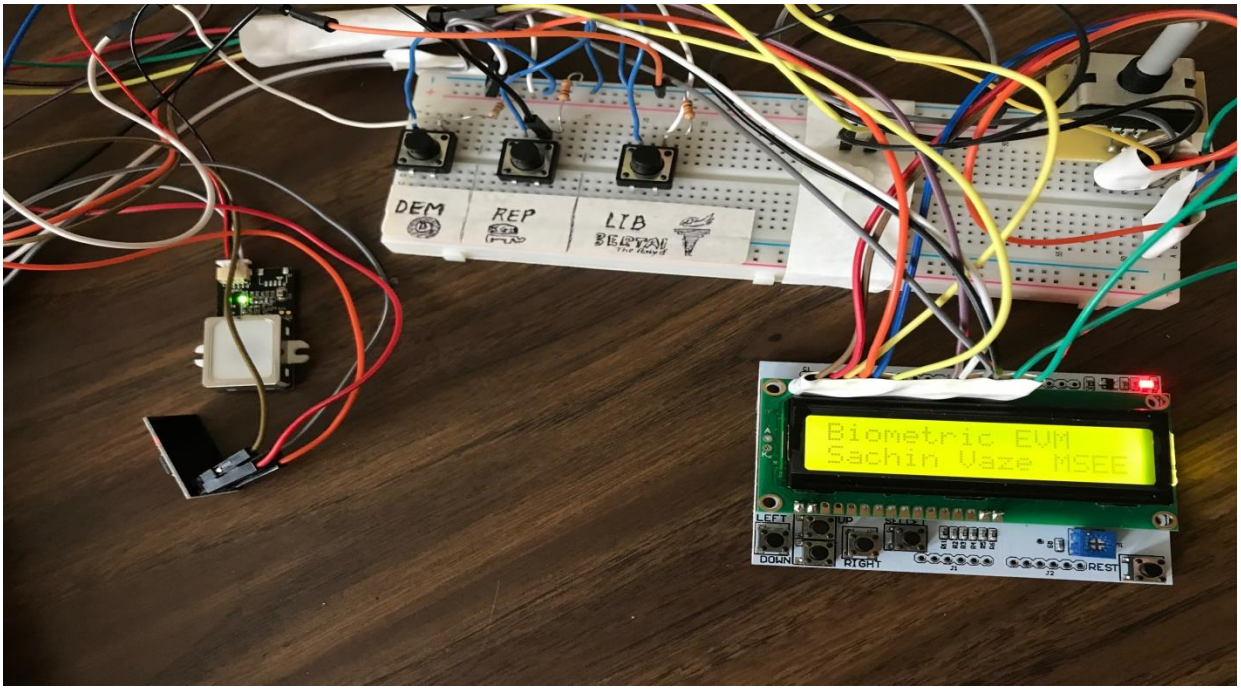
- Initially I tried the system with beaglebone. I was successful in establishing the connectivity, libraries' installation and SSH terminal set up to activate the command line interface for it. I could also make the LCD system work with the use of a library from Adafruit. The fingerprint sensor libraries were also successfully put in. I was going to use C &/or Python languages to do the programming part. But I found it particularly difficult in establishing a communication between any GUI and microcontroller. Also its hardware was very delicate and I could have used a BBB shield to guard my board from frying.
- Linux based command line coding is very useful and should be practiced more.
- Major next step was to accomplish cloud storage using current Arduino based system or with new BBB based system.
- Current system operates on limited data storage. More powerful system components like fingerprint modules and LCD modules can be used.

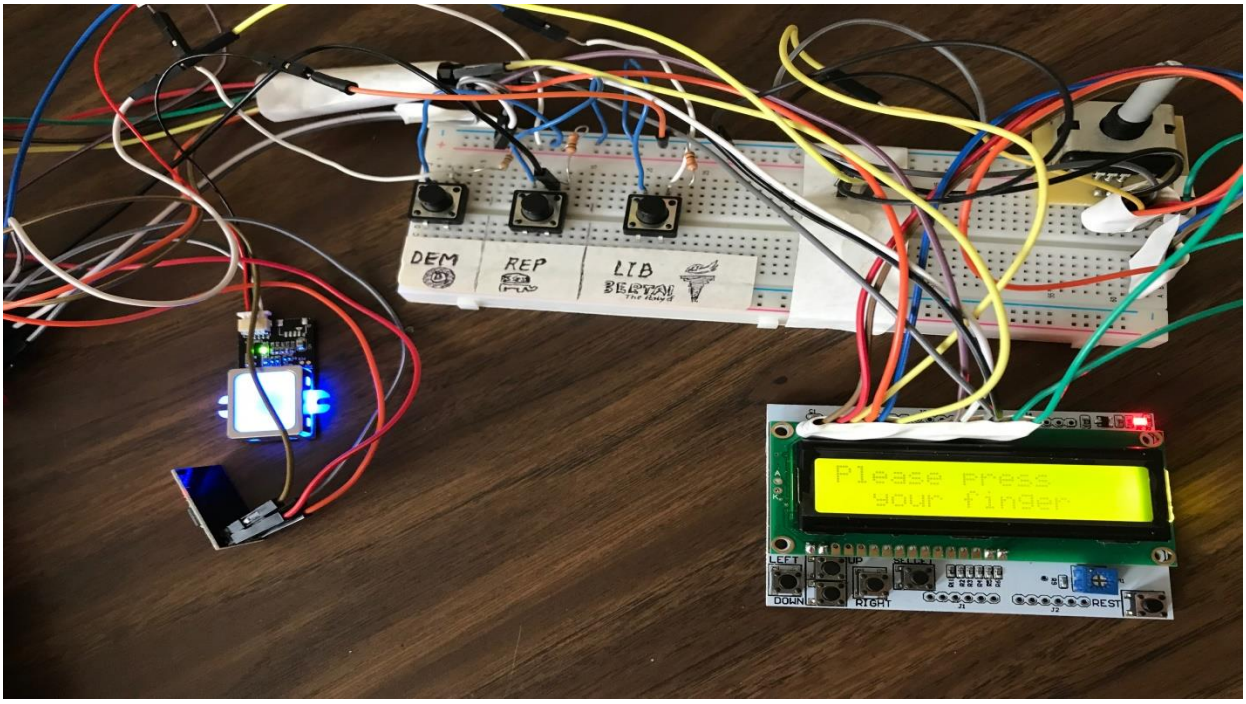
Result and Analysis

I have managed to take screen shots of my working project with corresponding inputs. The following functionalities displayed as shown:-

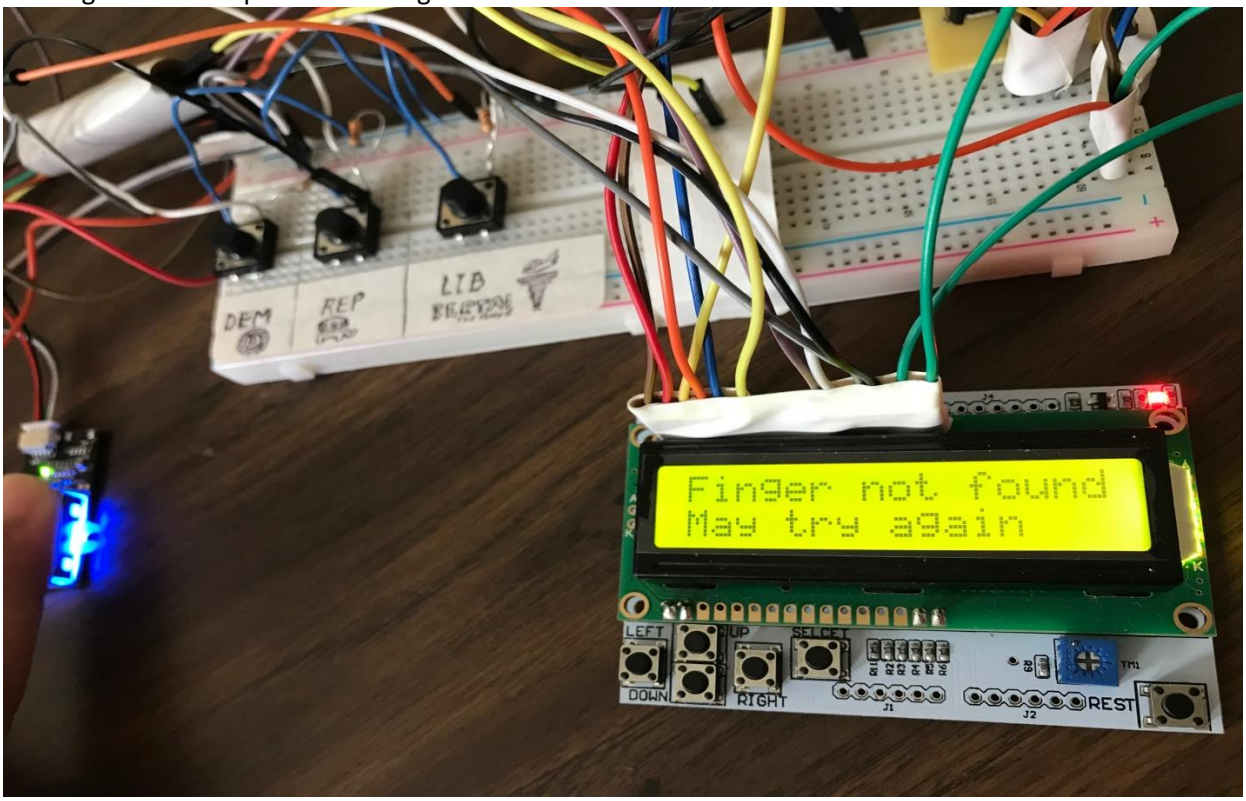
Upon Startup -



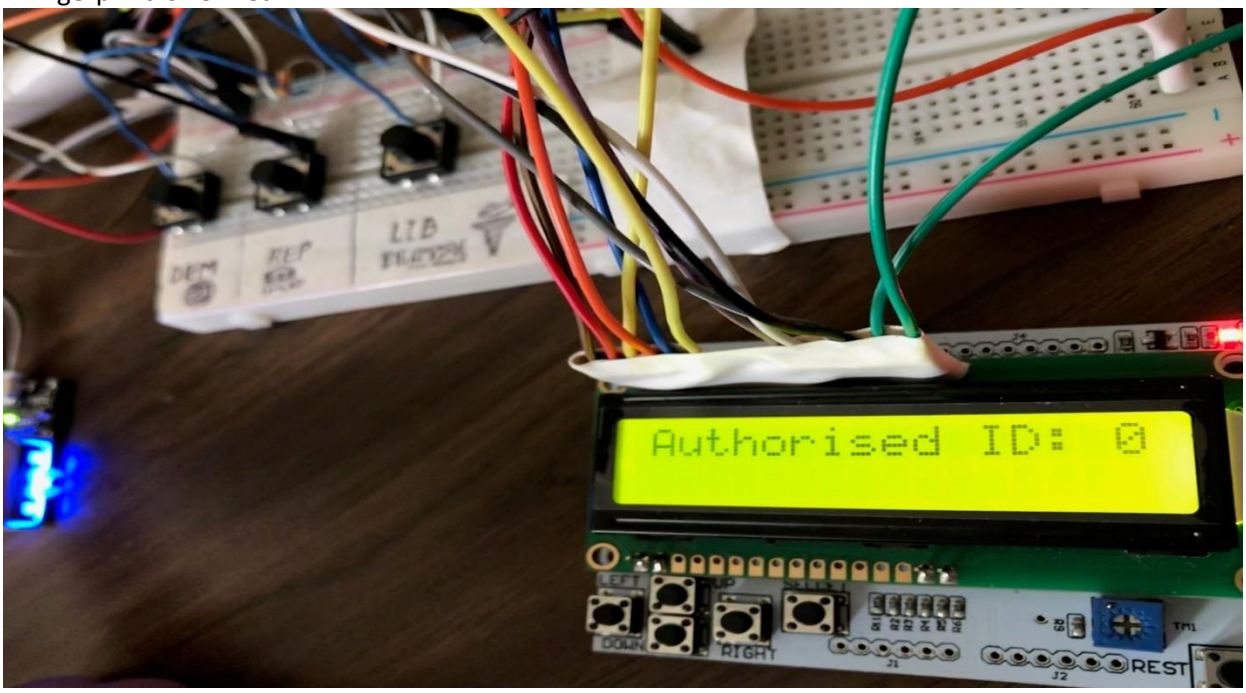


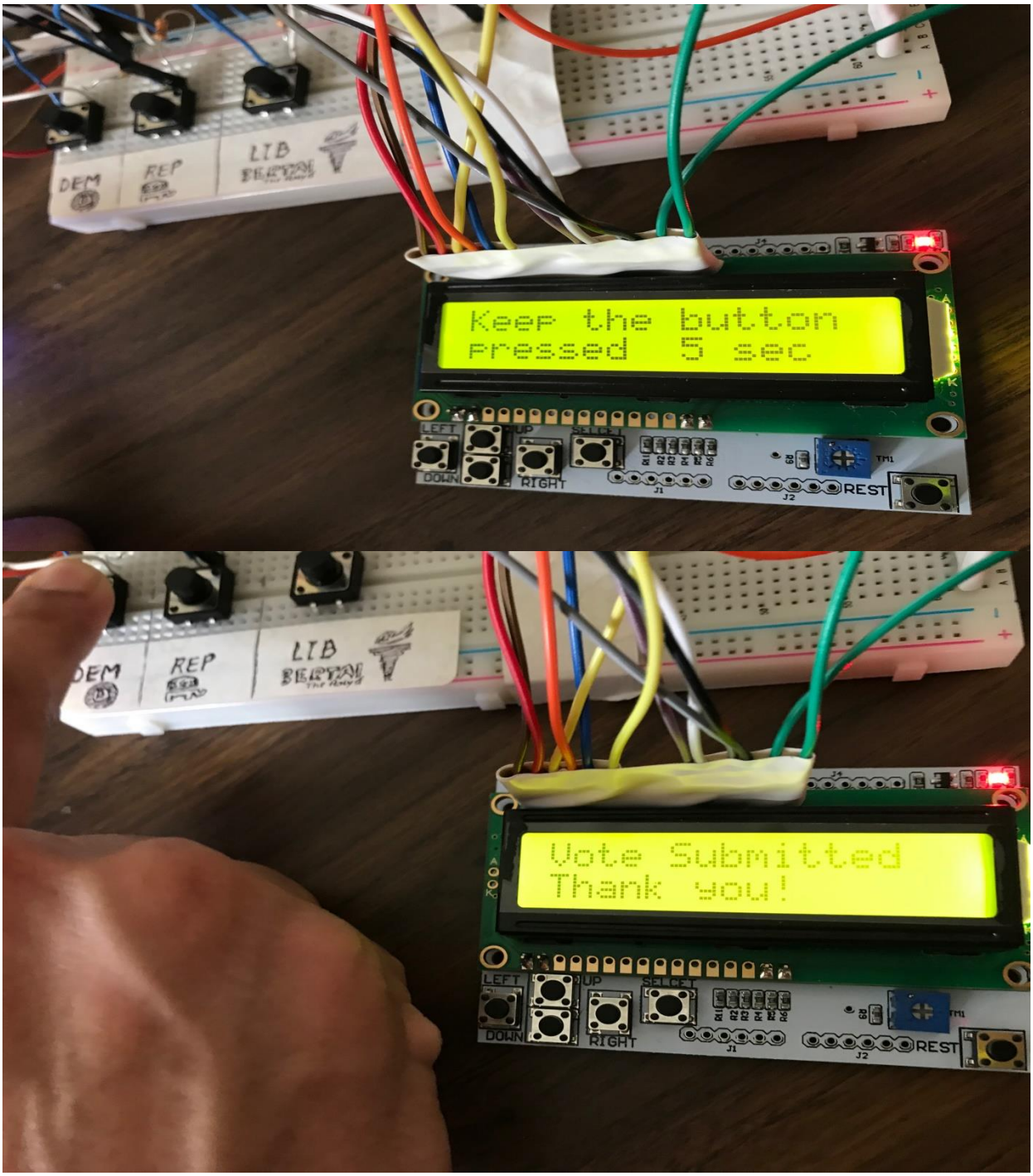


If unregistered user presses the finger –



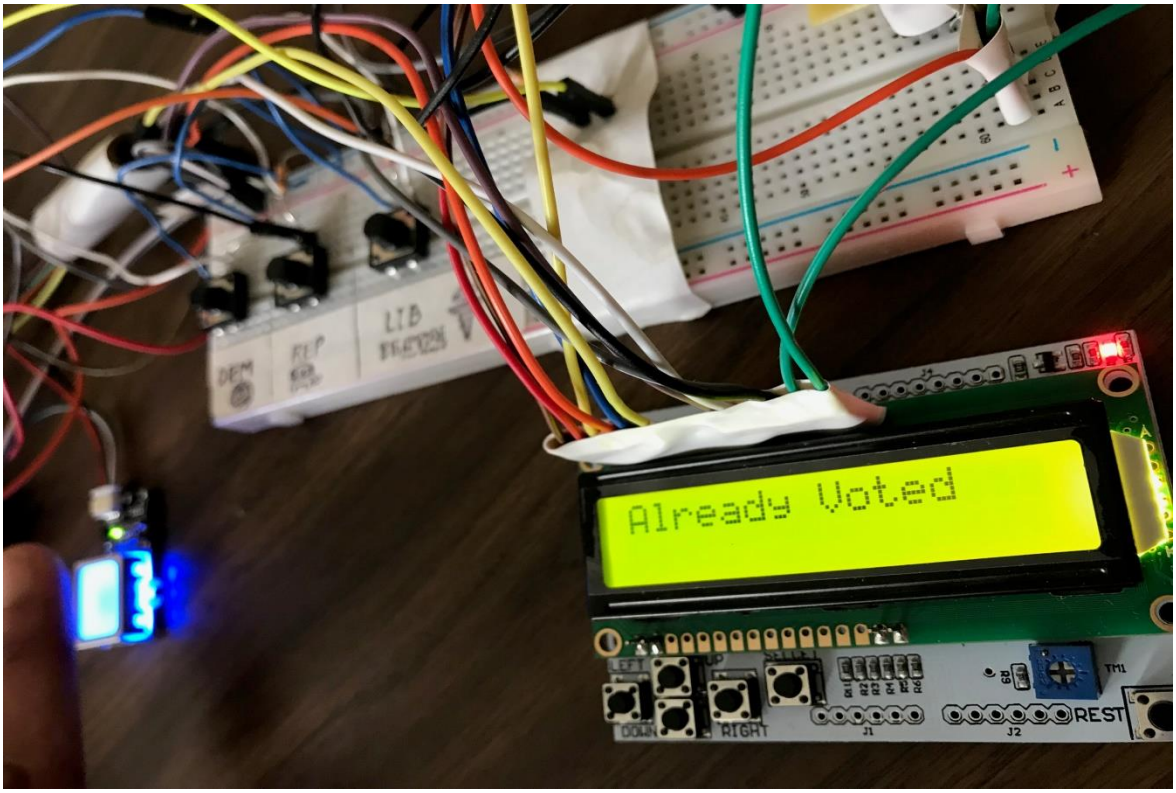
If fingerprint is verified -



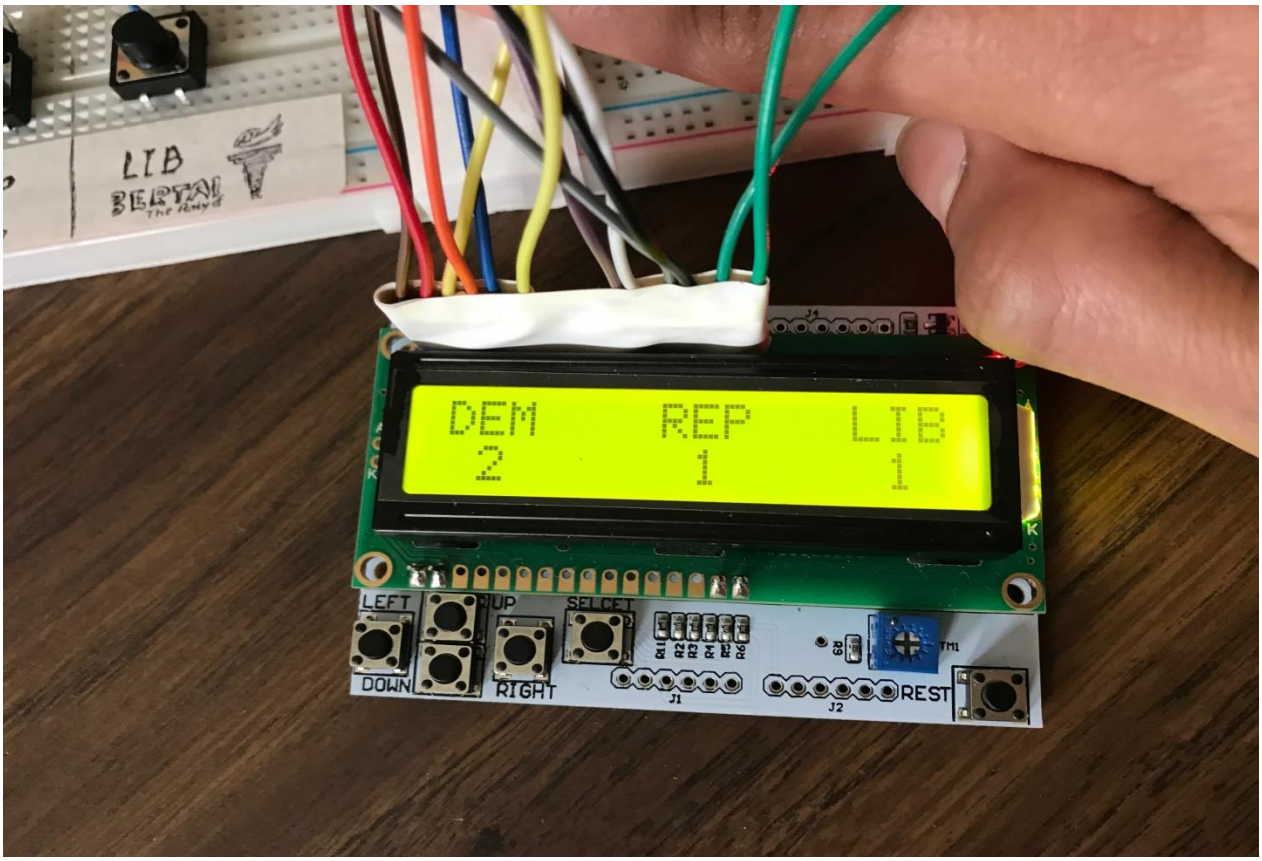


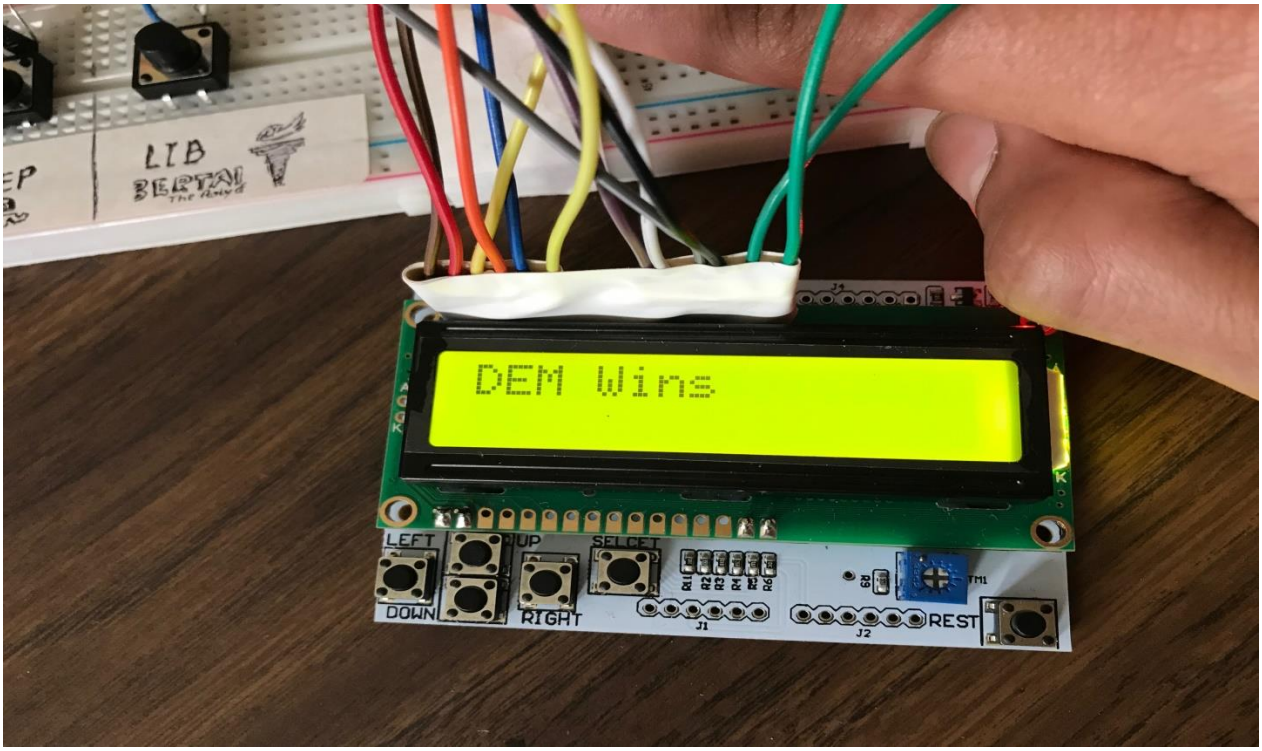
If new voter verifies the fingerprint (The ID gets authorized and the same procedure continues) –

If the same voter places the finger again –



Upon pressing the hidden ‘Result’ button (Only accessible to election authority) –


DEM REP LIB
2 1 1
LEFT
UP
SELCET
DOWN
RIGHT
REST
J1
J2
TM1


DEM Wins
LEFT
UP
SELCET
DOWN
RIGHT
REST
J1
J2
TM1

Future work

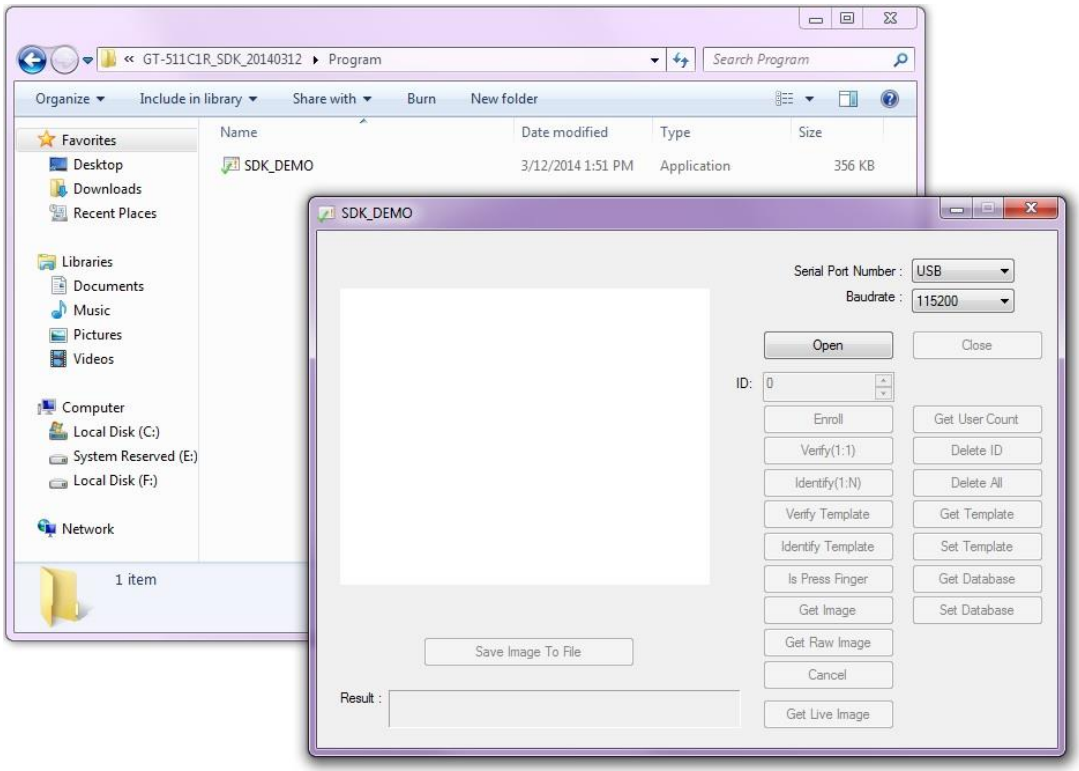
- With the addition of cloud storage, this system would become a perfect embedded design
- Once the cloud connectivity is attained, voter's information will be updated at real-time from anywhere and anytime.
- It would be possible to manage and process data from various centers by upgrading each of the systems to new changes in network routing and connectivity issues.
- Arduino proves out to be excellent electronics prototyping platform but its IDE doesn't support multi-tasking. So, development of this biometric EVM system on various platforms with various hardware as well as software solutions will be a topic of research in order to find the best implementable one. Beaglebone black, raspberry Pi are good possible options.
- Also Wi-Fi or web enabled Arduino boards could be better solution.
- Easier user applications using Android or iOS along with web features would boost the overall performance.

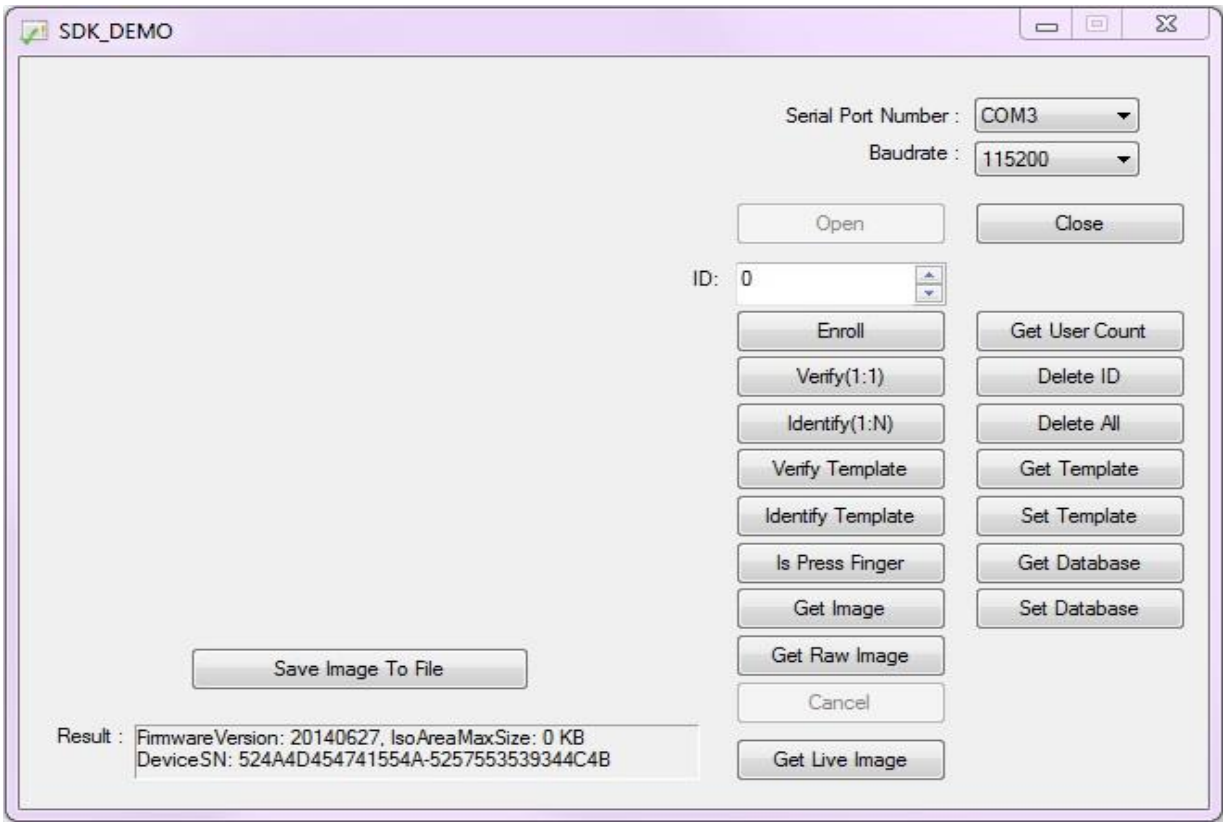
Conclusion/Summary

- The Biometrically secured Electronic Voting Machine successfully performs the voting process in an easy, quick, hassle-free and hack-proof manner.
- The accurate result is displayed as and when needed.
- The fingerprint modules are really helpful in designing simple yet highly secure and powerful embedded systems.
- Performance of the system can be improved with the use of more complex prototyping platforms like Raspberry Pi or Beaglebone black but the design complexity is also increased.
- Cloud connectivity and storage was one of the unsuccessful attempts but it can be demonstrated with some more efforts.
- Technically, conceptually and practically this embedded system design is a good example of high performance, high utility application in near future.

Appendix

SDK Demo software for fingerprint enrollment –





The Arduino C code for Fingerprint Sensor calibration –

```
#include "SoftwareSerial.h"
SoftwareSerial gtSerial(12,11); // Arduino RX, TX
void setup()
{
  Serial.begin(9600); //set up Arduino's hardware serial UART
  gtSerial.begin(9600); // for fingerprint scanner communication
 }
 byte rx_byte =0;
 void loop()
{
if(Serial.available()){
  // get byte from processing app and send to FPS
  rx_byte = Serial.read();
  gtSerial.write(rx_byte);
}
  // check for a byte from FPS
if(gtSerial.available()){
// get a byte from FPS and send it to processing app.
    rx_byte = gtSerial.read();
    Serial.write(rx_byte);
   }
   }
```

My Arduino C code for Biometric EVM –

```
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
#include "LiquidCrystal.h"
#include <stdbool.h>
// set up software serial pins for Arduino's w/ Atmega328P's
```

```
// FPS (TX) is connected to pin 4 (Arduino's Software RX)
// FPS (RX) is connected through a converter to pin 5 (Arduino's Software TX)
FPS_GT511C3 fps(12, 11); // (Arduino SS_RX = pin 12, Arduino SS_TX = pin 11)
LiquidCrystal lcd(2,3,4,5,6,7);
#define sw1 15
#define sw2 16
#define sw3 17
#define sw4 18
int vote1 = 0;
int vote2 = 0;
int vote3 = 0;
int usedID[20];
boolean flag;

/*If using another Arduino microcontroller, try commenting out line 53 and
uncommenting line 62 due to the limitations listed in the
library's note => https://www.arduino.cc/en/Reference/softwareSerial . Do
not forget to rewire the connection to the Arduino*/

// FPS (TX) is connected to pin 10 (Arduino's Software RX)
// FPS (RX) is connected through a converter to pin 11 (Arduino's Software TX)
//FPS_GT511C3 fps(10, 11); // (Arduino SS_RX = pin 10, Arduino SS_TX = pin 11)

void setup()
{
  usedID[0] = -4;
  flag = false;
  pinMode(sw1, INPUT);
  pinMode(sw2, INPUT);
```

```
 pinMode(sw3, INPUT);
 pinMode(sw4, INPUT);
 digitalWrite(sw1,HIGH);
 digitalWrite(sw2,HIGH);
 digitalWrite(sw3,HIGH);
 digitalWrite(sw4,HIGH);
 lcd.begin(16,2);
 Serial.begin(9600); //set up Arduino's hardware serial UART
 delay(1000);
 lcd.clear();
 lcd.setCursor(4,0);
 lcd.print("CSE 671");
 lcd.setCursor(1,2);
 lcd.print("Final Project");
 delay(2000);
 lcd.clear();
 lcd.print("Biometric EVM");
 lcd.setCursor(0,1);
 lcd.print("Sachin Vaze MSEE");
 delay(2000);
 lcd.clear();
 fps.Open();        //send serial command to initialize fps
 fps.SetLED(true);   //turn on LED so fps can see fingerprint
 }

void loop()
{
 delay(1500);
 if (digitalRead(sw4)==LOW)//Only voting authority has access to this button.
```

```
{
  Serial.println("Result switch pressed");

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("DEM");
  lcd.setCursor(1,1);
  lcd.print(vote1);
  lcd.setCursor(7,0);
  lcd.print("REP");
  lcd.setCursor(8,1);
  lcd.print(vote2);
  lcd.setCursor(13,0);
  lcd.print("LIB");
  lcd.setCursor(14,1);
  lcd.print(vote3);
  delay(3000);

  int vote=vote1+vote2+vote3;
  if(vote > 0)
  {
   if((vote1 > vote2 && vote1 > vote3))
   {
    lcd.clear();
    lcd.print("DEM Wins");
    delay(3000);
    lcd.clear();
   }
   else if((vote2 > vote1 && vote2 > vote3))
```

```
 {
  lcd.clear();
  lcd.print("REP Wins");
  delay(2000);
  lcd.clear();
 }
 else if((vote3 > vote1 && vote3 > vote2))
 {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("LIB Wins");
  delay(2000);
  lcd.clear();
 }
else
{
  lcd.clear();
  lcd.print("   Tie Up Or   ");
  lcd.setCursor(0,1);
  lcd.print("   No Result   ");
  delay(2000);
  lcd.clear();
 }

}
 else
 {
  lcd.clear();
  lcd.print("No Voting....");
```

```
    delay(2000);
    lcd.clear();
  }
  lcd.clear();
  //while(digitalRead(sw4) ==LOW);
 }
 else{
lcd.clear();
// Identify fingerprint test, if finger is pressed
if (fps.IsPressFinger())
{
  fps.CaptureFinger(false);
  int id = fps.Identify1_N();
      /*Note:  GT-521F52 can hold 3000 fingerprint templates
            GT-521F32 can hold 200 fingerprint templates
             GT-511C3 can hold 200 fingerprint templates.
            GT-511C1R can hold 20 fingerprint templates.
    Make sure to change the id depending on what
    model you are using */
  if (id <20) //<- change id value depending model you are using
  {//if the fingerprint matches, provide the matching template ID
    lcd.clear();
    Serial.print("Verified ID:");
    Serial.println(id);
    lcd.print("Authorised ID: ");
    lcd.print(id);
    delay(1000);

  if(usedID[id] == id && flag == false){
```

```
    // Electronic Voting begins
    lcd.clear();
    lcd.print("Already Voted");
    delay(2000);
    }
   else
   {
    Vote();
   }

    usedID[id] = id;
   }
   else
   {//if unable to recognize
    lcd.clear();
    lcd.print("Finger not found");
    lcd.setCursor(0,1);
    lcd.print("May try again");
    delay(1000);
    Serial.println("Ask voter to press finger again, if fails then unauthorized voter");
   }
  }
else
 {
   lcd.clear();
   lcd.print("Please press");
   lcd.setCursor(2,1);
   lcd.print("your finger");
   delay(1000);
```

```
    Serial.println("Ask voter to press finger");
  }
  delay(2000);
}
}

void Vote()
{
  lcd.clear();
  lcd.print("Place your vote");
  lcd.setCursor(7,1);
  lcd.print("NOW");
  delay(1800);
  lcd.clear();
  lcd.print("Keep the button");
  lcd.setCursor(0,1);
  lcd.print("pressed  5 sec");
  delay(4000);
  lcd.clear();
 if(digitalRead(sw1)== LOW){
    vote1++;
    flag = false;
    lcd.print("Vote Submitted");
    lcd.setCursor(0,1);
    lcd.print("Thank you!");
    delay(2000);
    lcd.clear();
    Serial.println("DEM count: ");
    Serial.println(vote1);
```

```
   //while(digitalRead(sw1)==LOW);
 }
  else if(digitalRead(sw2)==LOW){
   vote2++;
   flag = false;
   lcd.print("Vote Submitted");
   lcd.setCursor(0,1);
   lcd.print("Thank you!");
   Serial.print("REP count: ");
   Serial.println(vote2);
   //while(digitalRead(sw2)== LOW);
   }
 else if (digitalRead(sw3)==LOW){
    vote3++;
    flag = false;
    lcd.print("Vote Submitted");
    lcd.setCursor(0,1);
    lcd.print("Thank you!");
    Serial.println("LIB count");
    Serial.println(vote3);
    //while(digitalRead(sw3)== LOW);
    }
   else
    {

    flag = true;
    }
   return;
}
```

References and Bibliography

Information about Arduino and Arduino Nano –

https://learn.sparkfun.com/tutorials/what-is-an-arduino & https://store.arduino.cc/usa/arduino-nano

Project Idea links–

https://www.researchgate.net/publication/322789967_Fingerprint_Voting_System_Using_Arduino

https://circuitdigest.com/microcontroller-projects/fingerprint-based-biometric-voting-machine-arduino

https://www.youtube.com/watch?v=KLdU4Tgxi9w

https://www.hackster.io/sucheta/electronic-voting-machine-using-arduino-0d3da8

https://circuitdigest.com/microcontroller-projects/electronic-voting-machine-using-arduino

https://circuitdigest.com/microcontroller-projects/raspberry-pi-electronic-voting-machine

Links to find details about LCD display –

https://www.sparkfun.com/products/13293

http://linksprite.com/wiki/index.php5?title=16_X_2_LCD_Keypad_Shield_for_Arduino_V2

SDK Demo software –

https://cdn.sparkfun.com/datasheets/Sensors/Biometric/GT-511C1R_SDK_20140312.zip

Details of fingerprint sensor –

https://learn.sparkfun.com/tutorials/fingerprint-scanner-hookup-guide'

Details of ESP8266 – https://nurdspace.nl/ESP8266

Link for interfacing ESP8266 chip with Arduino Nano –

https://blogs.msdn.microsoft.com/abhinaba/2016/01/23/esp8266-wifi-with-arduino-uno-and-nano/

Arduino cloud platform – http://cloudino.io/

Google Firebase Real-time database –

https://firebase.google.com/

All about Beaglebone black –

https://beagleboard.org/black and http://derekmolloy.ie/tag/beaglebone-black/

Special Thanks to – Professor Ehat Ercanli (Teacher for CSE 671)