We all are aware of the existing electronic voting machine where the user has to press a button to cast the vote. But these machines have been criticized for tempering since the beginning. So the government is planning to introduce a fingerprint-based voting machine where users can cast the vote based on his/her fingerprint impression. This system will not only remove the possibility of duplicate votes but also prevent any kind of manipulation.



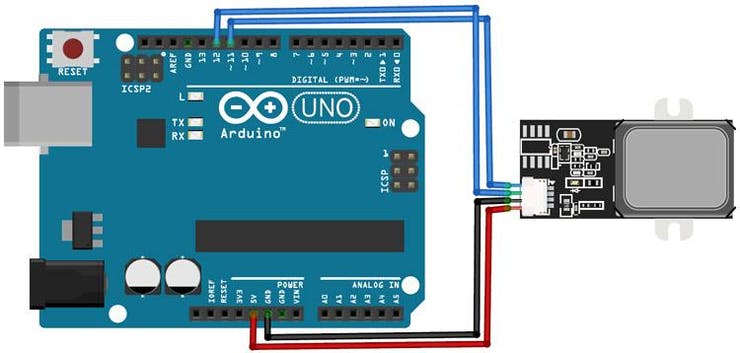
So in this project, we are going to build a prototype of a **Biometric voting machine using the Arduino Uno**, TFT display, and Fingerprint Sensor.We previously used R305 Fingerprint Sensor with NodeMCU to build a [Biometric based Attendance System](https://iotdesignpro.com/projects/iot-based-smart-attendance-system-project-using-nodemcu)but here we will use the advanced GT-511C3 fingerprint sensor with Arduino.

**Components Required to Build Biometric Voting Machine**

* [Arduino Uno](https://quartzcomponents.com/products/arduino-uno)
* [2.4” TFT LCD Display Shield](https://quartzcomponents.com/products/2-4-inch-tft-lcd-display-module-for-arduino-uno?_pos=1&_sid=36426caf1&_ss=r)
* [GT-511C3 Fingerprint Sensor](https://quartzcomponents.com/products/gt-511c3-fingerprint-scanner-module?_pos=1&_sid=e9c7440d8&_ss=r)

This 2.4 inch TFT display is previously used with Arduino to build [IoT Based Restaurant Menu Ordering System.](https://iotdesignpro.com/projects/iot-based-restaurant-menu-ordering-system)

**Circuit Diagram for Biometric Voting Machine Using Arduino**



Circuit Diagram for this project is very simple as we are only connecting the TFT display and fingerprint sensor module with Arduino Uno. VCC and GND pins of the fingerprint sensor are connected to 5V and GND pins of Arduino while TX and RX pins are connected to digital pin 11 & 12 of Arduino Uno.

The 2.4” TFT LCD screen is an Arduino Shield and can be directly mounted on Arduino Uno, as shown in the below image. TFT display has 28 pins that perfectly fit into Arduino Uno, so I had to solder the fingerprint sensor on the backside of Arduino.

**Source Code and Step by Step Code Explanation**

The complete code for this Fingerprint Voting System Project using Arduino is given at the end of the article; here we are explaining some important functions of the code.

The code uses the SPFD5408, Software Serial, and *FPS\_GT511C3* libraries. SPFD5408 library is the modified version of the original Adafruit Library. These library files can be downloaded from the links given below:

* [SPFD5408 Library](https://github.com/JoaoLopesF/SPFD5408)
* [Software Serial](https://github.com/PaulStoffregen/SoftwareSerial)
* [FPS\_GT511C3](https://github.com/sparkfun/Fingerprint_Scanner-TTL)

After including the libraries and defining some important parameters, we can get into the programming part. There are three sections involved in this program. One is creating a UI of a voting machine, second is getting the touch-points for buttons & detecting the buttons based on the touch and finally calculating the results and save them into Arduino’s memory.

**1. Creating UI:**

I have created a simple UI with three buttons and the name of the project. The TFT display library allows you to draw Lines, Rectangle, Circles, Chars, Strings and a lot more of any preferred color and size. Here two rectangular buttons are created using fillRoundRect and drawRoundRect functions. Syntax for tft.drawRoundRect function is given below:

*tft.drawRoundRect(int16\_t x0, int16\_t y0, int16\_t w, int16\_t h, int16\_t radius, uint16\_t color)*

*Where:*

*x0= X co-ordinate of the starting point of rectangular*

*y0= Y coordinate of the starting point of rectangular*

*w = Width of the rectangular*

*h = Height of the Rectangular*

*radius= Radius of the round corner*

*color = Colour of the Rect.*

*void drawHome()*

*{*

*tft.fillScreen(WHITE);*

*tft.drawRoundRect(0, 0, 319, 240, 8, WHITE); //Page border*

*tft.fillRoundRect(10, 70, 220, 50, 8, GOLD);*

*tft.drawRoundRect(10, 70, 220, 50, 8, WHITE); //Vote*

*tft.fillRoundRect(10, 160, 220, 50, 8, GOLD);*

*tft.drawRoundRect(10, 160, 220, 50, 8, WHITE); //Enroll*

*tft.fillRoundRect(10, 250, 220, 50, 8, GOLD); //Result*

*tft.drawRoundRect(10, 250, 220, 50, 8, WHITE);*

*tft.setCursor(65, 5);*

*tft.setTextSize(3);*

*tft.setTextColor(CYAN);*

*tft.print("Voting");*

*tft.setCursor(57, 29);*

*tft.print("Machine");*

*tft.setTextSize(3);*

*tft.setTextColor(WHITE);*

*tft.setCursor(25, 82);*

*tft.print("Candidate 1");*

*tft.setCursor(25, 172);*

*tft.print("Candidate 2");*

*tft.setCursor(25, 262);*

*tft.print("Candidate 3");*

*}*

**2. Getting the Touch-Points and Detecting Buttons:**

Now in the second section of the code, we will detect the button touchpoints and then use these points to predict the button. ts.getPoint() function is used to detect the user touch on TFT display. ts.getPoint gives the Raw ADC values for the touched area. These RAW ADC values are then converted to Pixel Coordinates using the map function.

*TSPoint p = ts.getPoint();*

*if (p.z > ts.pressureThreshhold)*

*{*

*p.x = map(p.x, TS\_MAXX, TS\_MINX, 0, 320);*

*p.y = map(p.y, TS\_MAXY, TS\_MINY, 0, 240);*

*//Serial.print("X:");*

*//Serial.print(p.x);*

*//Serial.print("Y:");*

*//Serial.print(p.y);*

Now, since we know the X and Y coordinates for each button, we can predict where the user has touched by using the ‘if’ statement.

*if (p.x > 70 && p.x < 120 && p.y > 10 && p.y < 220 && p.z > MINPRESSURE && p.z < MAXPRESSURE)*

*{*

*Serial.println("Candidate 1");*

When a voter presses the candidate button, he will be asked to scan the finger on the fingerprint sensor. If finger ID is authorized then the voter is allowed to vote. If any Non-registered user wants to vote then the fingerprint module will not detect its ID into the system and the display will show ‘Sorry You Can’t Vote ’.

*if (fps.IsPressFinger())*

*{*

*fps.CaptureFinger(false);*

*int id = fps.Identify1\_N();*

*if (id <200)*

*{*

*msg = "Candidate 1";*

*vote1++;*

*EEPROM.write(0, vote1);*

*tft.setCursor(42, 170);*

*tft.print("Thank You");*

*delay(3000);*

*drawHome();*

**3. Result:**

The last step is getting the vote count from EEPROM memory and comparing the votes of all three candidates. A Candidate with the highest votes wins. The result can only be accessed from the serial monitor and will not be displayed on the TFT screen.

*vote1=EEPROM.read(0);*

*vote2=EEPROM.read(1);*

*vote3=EEPROM.read(2);*

*if(vote)*

*{*

*if((vote1 > vote2 && vote1 > vote3))*

*{*

*Serial.print("Can1 Wins");*

*delay(2000);*

*}*

**Testing the Fingerprint Voting System Using Arduino**

To test the project, connect the Arduino Uno to the laptop and upload the code from [here](https://iotdesignpro.com/projects/fingerprint-based-biometric-voting-machine-using-arduino). Once the code is uploaded, the TFT display should display the candidate’s name. When someone taps on a candidate name, the machine will ask to scan the fingerprint scanner. If the fingerprint is valid, then the user vote will be counted, but in case, the pattern doesn’t match with the records of the database, access to cast a vote will be denied. The total number of votes for each candidate will be stored in EEPROM and a candidate having the highest number of votes will win.