BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 416 (January 2022) **C1**
Microprocessor and Embedded Systems Laboratory

Final Project Report

SMART EVM with IOT and FINGERPRINT

Evaluation Form:

STEP	DESCRIPTIO N	MAX	SCORE
1	Report (Format, Reference)	10	
2	Design Method and Complete Design (Hardware Implementation)	15	
3	Video Demonstration	10	
4	Novelty of Design	15	
5	Project Management and Cost Analysis	10	
6	Considerations to Public Health and Safety, Environment and Cultural and Societal Needs	10	
7	Assessment of Societal, Health, Safety, Legal and Cultural issues relevant to the solution	10	
8	Evaluation of the sustainability and impact of designed solution in societal and environmental contexts	10	
9	Individual Contribution (Viva)	20	
10	Team work and Diversity	10	
	TOTAL	120	

Signature of Evaluator: ______________________________

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and Write your name in your own handwriting. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.*

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Full Name: SAMIN ZAWAD **Student ID: 1706136**	**Full Name: MAHADI HASSAN** **Student ID: 1706137**
Full Name: SHAKHAWAT IQBAL OVI **Student ID: 1706138**	**Full Name: NAHID AHMED** **Student ID: 1706145**
Full Name: MD. ROKONUJJAMAN **Student ID: 1706150**	**Full Name: ISHMAM HASNAT** **Student ID: 1706153**

Table of Contents

1 Abstract

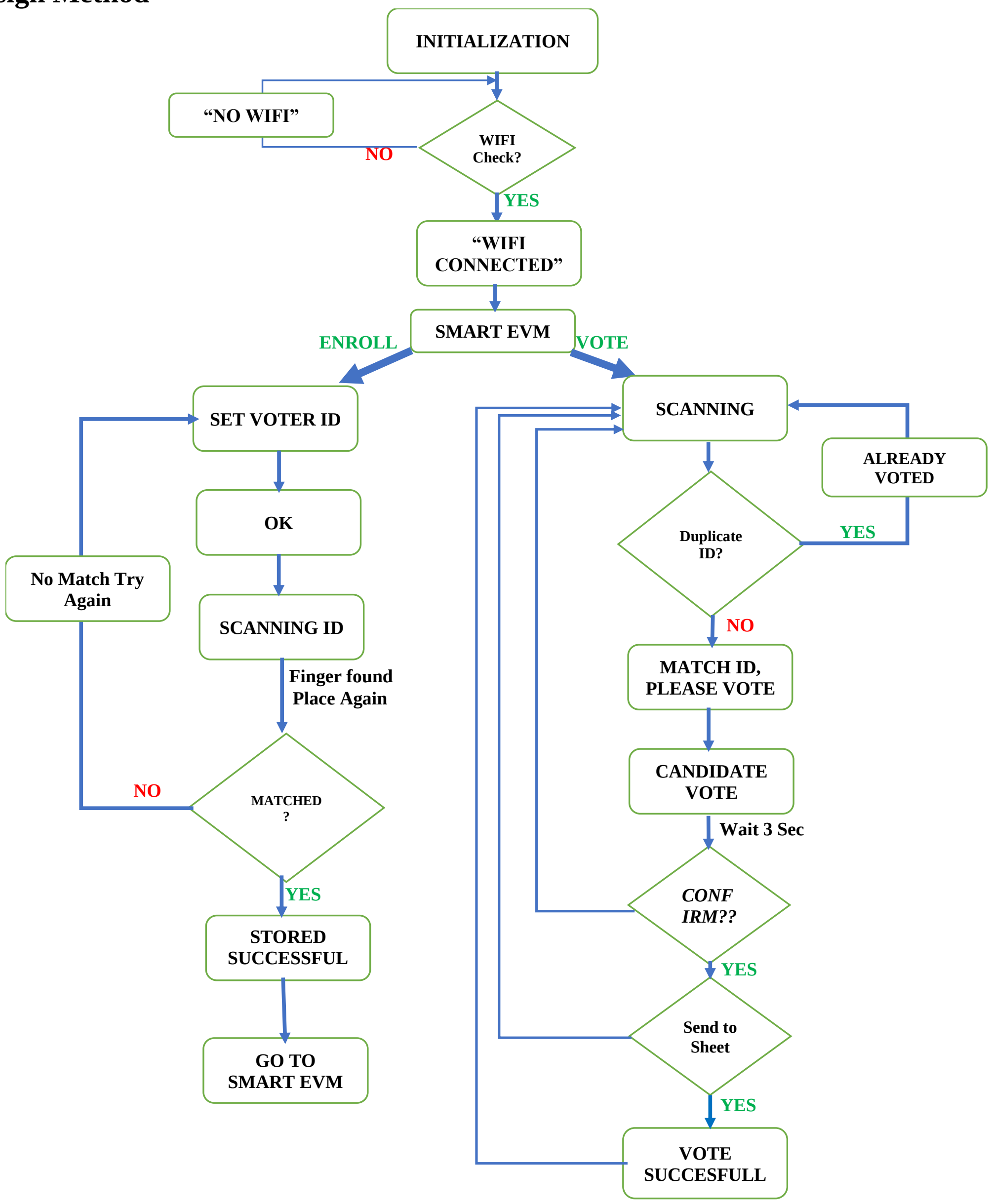
We implemented a IOT based electric voting machine that takes a voter's fingerprint, confirms identity, grants permission for Voting and updates the admin data sheet along with a public website. Server issue, Wi-Fi or internet unavailability detection with recovery voting system for robust and fair voting environment is ensured

2 Introduction

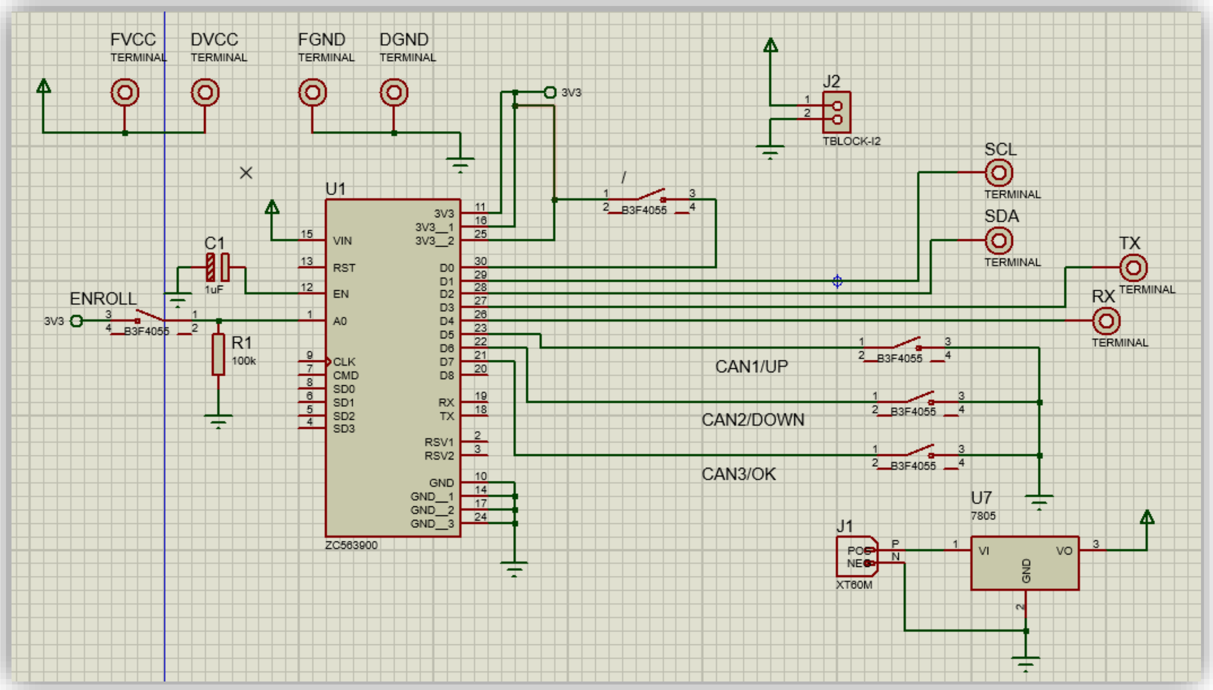
This EVM project is designed for 3 candidates. Voters can vote among 3 candidates of their consents. There are 2 parts of the project. In first part, voters can enroll their fingerprint in the database when admin permits. Fingerprints is saved in the database for future cross checking in time of voting. In 2nd part, voters can give vote after matching their fingerprint. A powerful code was uploaded in the microchip to implement features like 2nd time voting detecting, publishing vote count in the server, detection and recovery of server, Wi-Fi or internet malfunction, publishing result in a public access website etc.

3 Design

3.1 Design Method



3.2 Circuit Diagram



3.3 Full Source Code of Firmware

```
#include "TRIGGER_WIFI.h"
#include "TRIGGER_GOOGLESHEETS.h"
#include <Adafruit_Fingerprint.h>
//#include<ESP8266WiFi.h>
#include <LCD_I2C.h>
LCD_I2C lcd(0x27, 16, 2);
SoftwareSerial mySerial(D3, D4);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

char ssid[] = "ASUS";    // your network SSID (name)
char pass[] = "home1234"; // your network password

int votingdone[50];
int sizev = 0;
int buttonState = 0;
int voteflag = 0;  //voteflag = 1 implies vote was not successful
uint8_t id, count;


/*Google Sheets Definations*/
char column_name_in_sheets[ ][6] = {"pb1", "pb2", "pb3"};
String Sheets_GAS_ID = "AKfycbwgsODCxfj0bYvSgEB3kumGH3P1KVFZf16_Kflzji2zMBGH0gGAeP5zS1FzvHVrZBb4";
int No_of_Parameters = 3;

/*EVM Start Here*/
void setup()
{
  Serial.begin(115200);
  lcd.begin();
  lcd.backlight();
  lcd.clear();
  bool j;
  while (1)
  {
    j = WIFI_Connect(ssid, pass);
    if (j == false) {
      lcd.clear();
      lcd.print("No Wifi found");
      delay(500);
    }
    else {
      Serial.println("Wifi Connected");
      break;
    }
  }
  Google_Sheets_Init(column_name_in_sheets, Sheets_GAS_ID, No_of_Parameters );
  pinMode(D5, INPUT_PULLUP);      // Can1 / UP Dual Function Button
  pinMode(D6, INPUT_PULLUP);      // Can2 / Down Dual Function Button
  pinMode(D7, INPUT_PULLUP);      // Can3 / OK Dual Function Button
  pinMode(D0, INPUT_PULLDOWN_16); //Confirm Button

  lcd.setCursor(0, 0);  lcd.print("SMART EVM with");
  lcd.setCursor(0, 1);  lcd.print("IOT&Fingerprint");
  delay(2000);

  finger.begin(57600);
  delay(5);
  if (finger.verifyPassword())
  {
    Serial.println("Found Module!");
  }
  else
  {
    Serial.println("Did not find fingerprint sensor");
    lcd.clear();
    lcd.print("Fingerprint module not Found");
    lcd.setCursor(0, 1);  lcd.print("Check Connections");
    while (1) {
      delay(1);
    }
  }
  finger.getTemplateCount();
  if (finger.templateCount == 0)
  {
    Serial.print("Sensor doesn't contain any fingerprint data. Please enroll fingeprints.");
  }
  else
  {
    Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
  }
  int i;
  for (i = 0; i < 50; i++)
  {
    votingdone[i] = -10;
  }
}

void loop()
{
  lcd.clear();
  lcd.setCursor(0, 0);  lcd.print("Vote | Enroll?");
  lcd.setCursor(0, 1);  lcd.print("Press Button");
  delay(500);
  Serial.println("Vote | Enroll?");
  if (analogRead(A0) > 1000)        //enters enroll mode
    enrollID();
  else if (digitalRead(D0))
  {
    while (1)
    {
      delay(1);
      int result = getFingerprintIDez();
      int flag = 0, i;
      delay(50);
      Serial.println("Scanning");
      lcd.clear();
      lcd.setCursor(0, 0);  lcd.print("Scanning");

      if (result >= 0)
      {
        Serial.print("Matched FPID is:  "); Serial.println(result);
        flag = 0;
        for (i = 0; i < sizev; i++)
        {
          if (result == votingdone[i])
          {
            flag = 1;
            Serial.println("Duplicate Voter");
            lcd.clear();
            lcd.print("DuplicateID #"); lcd.print(result);
            lcd.setCursor(0, 1);  lcd.print("Already Voted");
            delay(5000);
            break;
          }
        }
        if (flag == 0)
        {
          lcd.clear();
          lcd.setCursor(0, 0);  lcd.print("MatchedID #"); lcd.print(result);
          lcd.setCursor(0, 1);  lcd.print("Please Vote");
          buttonpress();  //Voting Function
          //      Serial.print("VF = ");
          //      Serial.println(voteflag);
          if (!voteflag) {
            votingdone[sizev] = result;
            sizev += 1;
          }
          voteflag = 0;  //pore
        }
      }
      if (analogRead(A0) > 1000)        //Exits Vote Mode
        break;
    }
  }
}

/*Enroll Main Function*/
void enrollID()
{
  while (1)
  {
    Serial.println("Ready to enroll a fingerprint!");
    Serial.println("Please type in the ID # (from 1 to 127) you want to save this finger as...");
    lcd.clear();
    lcd.setCursor(0, 0);  lcd.print("Enter New FP-ID");
    lcd.setCursor(0, 1);  lcd.print("1");
    delay(500);
    id = mycount();
    if (id == 0)
      return;
    Serial.print("Enrolling ID #");
    Serial.println(id);
    if ((getFingerprintEnroll()) != FINGERPRINT_ENROLLMISMATCH)
      return;
  }
}
/* ~Enroll Main Function */

/* Enroll by Push Button */
uint8_t mycount(void) {
  count = 1;
  Serial.print("Enter New FP-ID");
  while (1) {
    delay(1);

    if (digitalRead(D5) == 0) // if Up button is pressed
    {
      count++;
      if (count > 50)
        count = 1;
      Serial.print("+++>"); Serial.println(count);
      lcd.clear();
      lcd.setCursor(0, 0);  lcd.print("Enter New FP-ID");
      lcd.setCursor(0, 1);  lcd.print(count);
      delay(500);
    }
    else if (digitalRead(D6) == 0)
    {
      count--;
      if (count < 1)
        count = 50;
      Serial.print("--->"); Serial.println(count);
      lcd.clear();
      lcd.setCursor(0, 0);  lcd.print("Enter New FP-ID");
      lcd.setCursor(0, 1);  lcd.print(count);
      delay(500);
    }
    else if (digitalRead(D7) == 0)
    {
      Serial.print("Scanning for ID: ");
      Serial.println(count);
      lcd.clear();
      lcd.setCursor(0, 0);  lcd.print("Scanning for ");
      lcd.setCursor(0, 1);  lcd.print("ID:");
      lcd.setCursor(4, 1);  lcd.print(count);
      delay(500);
      return count;
    }
  }
}

/*~Enroll ID by Push Button */

/*Enroll Process Start*/
uint8_t getFingerprintEnroll() {
  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
      case FINGERPRINT_OK:
        Serial.println("Image taken");
        break;
      case FINGERPRINT_NOFINGER:
        Serial.println(".");
        break;
      case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        break;
      case FINGERPRINT_IMAGEFAIL:
        Serial.println("Imaging error");
        break;
      default:
        Serial.println("Unknown error");
        break;
    }
  }

  // OK success!

  p = finger.image2Tz(1);
  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;
  }

  Serial.println("Remove finger");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Print Taken");
  lcd.setCursor(0, 1);
  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.setCursor(0, 0); lcd.print("Place Finger ");
  lcd.setCursor(0, 1); lcd.print("Again");
  //  delay(500);
  while (p != FINGERPRINT_OK) {
    if (digitalRead(D0))
      break;
    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");
        break;
    }
  }

  // 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!");

    //delay(3000); //edit
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    lcd.clear();
    lcd.setCursor(0, 0);  lcd.print("No Match");
    lcd.setCursor(0, 1);  lcd.print("Fail, Try Again");
    delay(1000); //edit
    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 ID:");
    lcd.clear();
    lcd.setCursor(0, 0);  lcd.print("Saved ID:");
    lcd.setCursor(12, 0); lcd.print(id);
    lcd.setCursor(0, 1);  lcd.print("Success");
    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;
  }
  return true;
}
/*Enroll Process End*/

/*Finger Matching Start*/
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)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  //  lcd.clear();
  //  lcd.print("Found ID #"); lcd.print(finger.fingerID);
  //  lcd.setCursor(0, 1);
  //  lcd.print("Press To Vote");

  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return finger.fingerID;
}
/*Finger Matching End*/

/*Send of Sheets Start*/
void buttonpress() {
  unsigned long canceldelay = 3000;
  int pb1, pb2, pb3;
  int data1 = 0, data2 = 0, data3 = 0;
  bool success = false;//pore
  while ((!data1) && (!data2) && (!data3)) {
    delay(1);
    pb1 = digitalRead(D5);
    pb2 = digitalRead(D6);
    pb3 = digitalRead(D7);

    data1 = pb1 ? 0 : 1;  //flips states
    data2 = pb2 ? 0 : 1;
    data3 = pb3 ? 0 : 1;

    if ((data1 == 1) && (data2 == 0) && (data3 == 0)) {

      lcd.clear();
      lcd.print("Can1 , Sure?");
      delay(1000);
      buttonState = digitalRead(D0); //Low if not pressed, High if pressed
      unsigned long entrypoint = millis();
      while (!buttonState)
      {
        delay(1);
        buttonState = digitalRead(D0);
        //Serial.println(buttonState);

        if (((millis() - entrypoint) > canceldelay)) {
          voteflag = 1;
          lcd.clear();
          lcd.print("NoVote-TryAgain");
          Serial.println("NoVote-TryAgain");
          delay(2000);
          break;
        }
        else if (((millis() - entrypoint) > 2000))
        {
          // lcd.clear();
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("1 Sec");
          Serial.println("Time Remaining 1 Sec");
        }
        else if (((millis() - entrypoint) > 1000))
        {
          // lcd.clear();
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("2 Sec");
          Serial.println("Time Remaining 2 Sec");
        }
        else if (((millis() - entrypoint) > 0))
        {

          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("3 Sec");
          Serial.println("Time Remaining 3 Sec");
        }

      }
      if (buttonState) {
        lcd.clear();
        lcd.setCursor(0, 0);  lcd.print("Can1 Vote");
        success = Data_to_Sheets(No_of_Parameters, (float) data1, (float) data2, (float) data3);
        Serial.println(success);
        if (success == true) {
          lcd.setCursor(0, 1);  lcd.print("Vote Successful!");
          Serial.println("Vote Successful!");
          delay(3000);
        }
        else {
          lcd.setCursor(0, 1);  lcd.print("Vote Failed!");
          Serial.println("Vote Failed!");
          delay(3000);
          voteflag = 1;
          return;
        }
      }
    }
    else if ((data1 == 0) && (data2 == 1) && (data3 == 0)) {
      lcd.clear();
      lcd.print("Can2 , Sure?");
      delay(1000);
      buttonState = digitalRead(D0); //Low if not pressed, High if pressed
      unsigned long entrypoint = millis();
      while (!buttonState)
      {
        delay(1);
        buttonState = digitalRead(D0);

        if (((millis() - entrypoint) > canceldelay)) {
          voteflag = 1;
          lcd.clear();
          lcd.print("NoVote-TryAgain");
          Serial.println("NoVote-TryAgain");
          delay(2000);
          break;
        }
        else if (((millis() - entrypoint) > 2000))
        {
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("1 Sec");
          Serial.println("Time Remaining 1 Sec");
        }
        else if (((millis() - entrypoint) > 1000))
        {
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("2 Sec");
          Serial.println("Time Remaining 2 Sec");
        }
        else if (((millis() - entrypoint) > 0))
        {
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("3 Sec");
          Serial.println("Time Remaining 3 Sec");
        }
      }
      if (buttonState) {
        lcd.clear();
        lcd.setCursor(0, 0);  lcd.print("Can2 Vote");
        success = Data_to_Sheets(No_of_Parameters, (float) data1, (float) data2, (float) data3);
        if (success == true) {
          lcd.setCursor(0, 1);  lcd.print("Vote Successful!");
          Serial.println("Vote Successful!");
          delay(3000);
        }
        else {
          lcd.setCursor(0, 1);  lcd.print("Vote Failed!");
          Serial.println("Vote Failed!");
          delay(3000);
          voteflag = 1;
          return;

        }
      }
    }
    else if ((data1 == 0) && (data2 == 0) && (data3 == 1)) {
      lcd.clear();
      lcd.print("Can3 , Sure?");
      delay(1000);
      buttonState = digitalRead(D0); //Low if not pressed, High if pressed
      unsigned long entrypoint = millis();
      while (!buttonState)
      {
        delay(1);
        buttonState = digitalRead(D0);

        if (((millis() - entrypoint) > canceldelay)) {
          voteflag = 1;
          lcd.clear();
          lcd.print("NoVote-TryAgain");
          Serial.println("NoVote-TryAgain");
          delay(2000);
          break;
        }
        else if (((millis() - entrypoint) > 2000))
        {
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("1 Sec");
          Serial.println("Time Remaining 1 Sec");
        }
        else if (((millis() - entrypoint) > 1000))
        {
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("2 Sec");
          Serial.println("Time Remaining 2 Sec");
        }
        else if (((millis() - entrypoint) > 0))
        {
          lcd.setCursor(0, 0);  lcd.print("Time Remaining:");
          lcd.setCursor(0, 1);  lcd.print("3 Sec");
          Serial.println("Time Remaining 3 Sec");
        }
      }
      if (buttonState) {
        lcd.clear();
        lcd.setCursor(0, 0);  lcd.print("Can3 Vote");
        success = Data_to_Sheets(No_of_Parameters, (float) data1, (float) data2, (float) data3);
        if (success == true) {
          lcd.setCursor(0, 1);  lcd.print("Vote Successful!");
          Serial.println("Vote Successful!");
          delay(3000);
        }
        else {
          lcd.setCursor(0, 1);  lcd.print("Vote Failed!");
          Serial.println("Vote Failed!");
          delay(3000);
          voteflag = 1;
          return;

        }
      }
    }
    if ((data1 == 1) || (data2 == 1) || (data3 == 1)) {
      delay(50);
      return;
    }
  }
}
/*Send of Sheets End*/
```

Table: Source Code for the main program

4 Implementation

4.1 PCB Layout

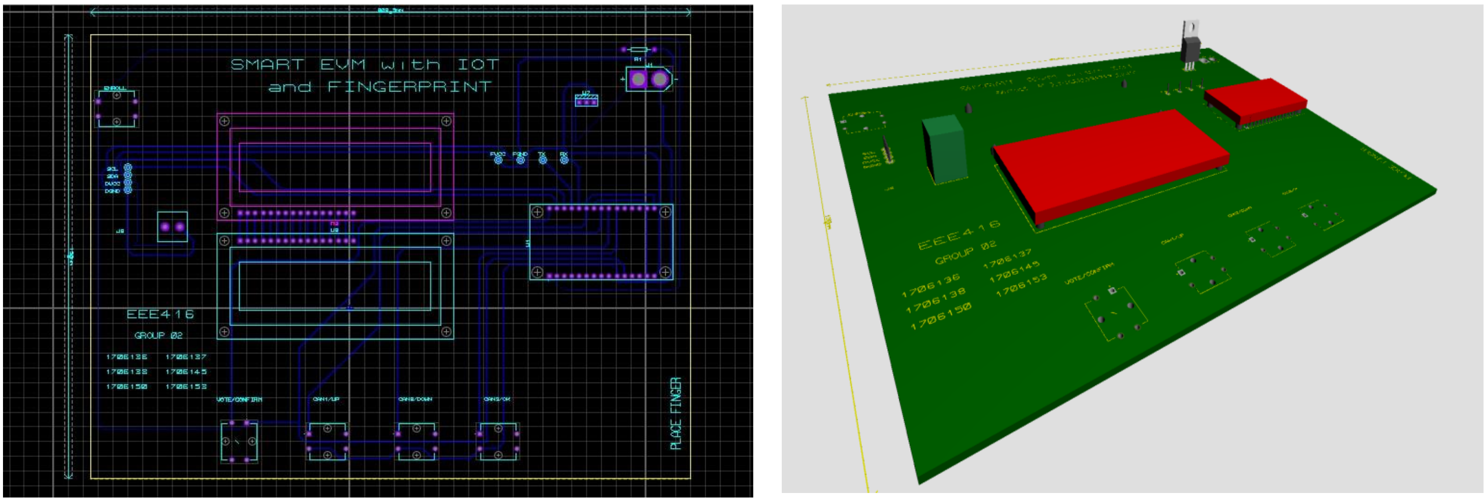


Figure 2: (Left) PCB Layout and (Right) Implementation of Design

Pin Type	Pin	Configuration	Connection Description
Digital Pins	D0	PULL-DOWN	Vote/Confirm push button is attached as a pull-down switch and become high after connecting to 3.3V pin when switch is pushed.
	D1	SCL	Connected to SCL (Serial clock) port of I2C module which synchronizes the transferred data
	D2	SDL	Connected to SDL (Serial Data) used to transfer data from I2C to other connected device.
	D3	TX	Connected to TX port of fingerprint sensor
	D4	RX	Connected to RX port of fingerprint sensor
	D5	PULL-UP	CAN1/UP button is connected as pull up
	D6	PULL-UP	CAN2/DOWN button is connected as pull up
	D7	PULL-UP	CAN3/OK button is connected as pull up
	D8	UNUSED	Boot fails if pull high
Analog Pin	A0	PULL-DOWN	NODEMCU ESP8266 devide 3.3V to 1024 level. It takes an analog input and convert this to the corresponding binary data. ENROLL button is attached to A0 pin and connected to ground as pull down to avoid floating condition.
Power	Vin	5 V	Direct 5V 2A DC adapter and 3S 11.1V LiPo 1200 mAH with 7805 Voltage regulator and capacitor. Capacitor is used to control the ripple of the DC current.

4.2 Results

Safe Voting Procedure with real time data monitor and surplus from any attempt of duplicate voting and ensures the transparency of voting procedure by uploading data to server

4.3 Github Link

4.4YouTube Link

https://www.youtube.com/watch?v=OSndbgKLA34

5 Design Analysis and Evaluation

5.1 Novelty

- Dual power backup.
- Real time update of vote & Update to website after a time interval.
- Dual purpose (Voting + Enrolling).
- Detect suspicious activities because of Real Time Monitoring
- ESP8266 is cheaper Microcontroller compared to ESP32
- No Arduino used.
- Voter’s privacy is maintained.
- Human Error in pushing candidates voting button is managed
- Vote error due to internet disruption is dealt with

5.2 Project Management and Cost Analysis

5.2.1 Bill of Materials

Product	**Model**	**Quantity**	**Unit Price(Tk)**	**Total(Tk)**
Fingerprint sensor	R307	1	1300	1300
WiFi Module	ESP 8266	1	200	200
Push Button		5	8	40
Wire		2	40	80
Omron Tact Switch	SWT- 111212	10	5	50
40*1 Female Headers	1FPH25B401	3	12	36
XT60 Connector’s Pair	CON- 10260	1	70	70
Multi Colors Cap for Tactile Push Switch	CAP-12121	10	2	20
Jia Long Xing 1200 mAh 11.1 V 3S 45C Li-Po Battery		1	1100	1100
2x16 LCD Display		1	250	250
I2C LCD adapter		1	90	90
L7805 IC		1	10	10
5V 2A AC to DC Adapter Power Supply Charger		1	240	240
PVC Plastic		1	200	200
PCB (5.5 Gis) 8’x 6’ + Processing Charge		1		680
Total				**4366**

5.2.2 Calculation of Per Unit Cost of Prototype

Per Unit Cost of prototype is **4366 Tk.**

5.2.3 Calculation of Per Unit Cost of Mass-Produced Unit

If mass raw elements are purchased, the cost of IC, and other equipment are reduced. By rough calculation and approximation, per unit cost of mass- produced unit is **2500 Tk.**

5.2.4 Timeline of Project Implementation

Week	**Timeline**
7th	Component Purchased
8th	Implementation of Fingerprint Sensor
9th	Schematic Design and Implementation of Circuit
10th	IoT Design and Implementation
11th	Coding in Arduino ide
12th	PCB design
13th	Soldering and Hardware Implementation in PCB
14th	Report writing and making Video Demonstration

5.3 Practical Considerations of the Design to Address Public Health and Safety, Environment, Cultural, and Societal Needs

5.3.1 Considerations to public health and safety

We didn’t use any kind of devices that is harmful to public health and safety. The used electronic devices work in low voltage. So, if any malfunction happens, there is no possibility to get shocked of personnel. Also, proper privacy is maintained of voting data and remains hidden. So, people can see only result of election in server.

5.3.2 Considerations to environment

There is no hazardous equipment used in our project that are harmful to environment. Where this smart EVM machine save the cost of simply printing and distributing the millions of ballots required to make an election happen in the ballot paper system. Rejecting the ballot paper system will help to improve our environment and reduce deforestation. We used DC battery source which reduces electricity cost.

5.3.3 Considerations to cultural and societal needs

The EVM system leads to strengthening of democratic institution and better representation of marginalized and vulnerable segments of society. Reducing electoral fraud, it has to make elected officials more accountable.

5.4 Assessment of the Impact of the Project on Societal, Health, Safety, Legal and Cultural Issues

5.4.1 Assessment of Societal Issues

It reduces the public gathering. People get known of this new device and aware of the digital electronic product.

5.4.2 Assessment of Health and Safety Issues

Public health and safety are most prior thing to use this device. we didn’t use any kind of hazardous material for this project. It can increase voter turnouts, integrity, and security and thus defends democracy.

5.4.3 Assessment of Legal Issues

The device is legal by the code of govt. rules. Also, a person can only vote one time and if any fraud comes to vote again, it detects the fraud. There is no chance of delaying the vote count, or changing the result under the cover of delayed announcement.

5.4.4 Assessment of Cultural Issue

The EVM framework leads to fortifying of equitable institution and way better representation of marginalized and powerless portions of society. Lessening discretionary extortion, it needs to make chosen authorities more responsible.

5.5 Evaluation of the Sustainability the and Impact of the Designed Solution in the Societal and Environmental Contexts

5.5.1 Evaluation of Sustainability

In our project, It is reusable and cost effective and ensure a faster and more accurate collation of votes. It reduces the total number of votes; since voters know when their choice is recorded. They are not allowed to choose more than one candidate.

5.5.2 Evaluation of Impact of Design in Societal Context

The EVM framework leads to fortifying of law-based institution and superior representation of marginalized and powerless fragments of society. Diminishing appointive extortion, it needs to make chosen authorities more responsible.

5.5.3 Evaluation of Impact of Design in Environmental Context

There's no perilous hardware utilized in our venture that are hurtful to environment. Where this keen EVM machine spare the fetched of basically printing and dispersing the millions of ballots required to form a decision happen within the vote paper framework. Dismissing the vote paper framework will offer assistant to move forward our environment and decrease deforestation. We utilized DC battery source which diminishes power taken a toll.

6 Reflection on Individual and Team work

6.1 Individual Contribution of Each Member

We have distributed the task such as fingerprint sensor, PCB design, Enclosure box making, Nodemcu working, hardware implementation and IoT based website design to individual member of team. Then we have conjoined all task together.

6.2 Mode of TeamWork

We have worked together in hall to conjoin the segments of project.

6.3 Diversity Statement of Team

Our team members are from power major, communication major background. So, all of our member are qualified in different skill which helped to stand this project.

6.4 Log Book of Project Implementation

Week	Timeline
7th	Component Purchased
8th	Implementation of Fingerprint Sensor
9th	Schematic Design and Implementation of Circuit
10th	IoT Design and Implementation
11th	Coding in Arduino IDE
12th	PCB design
13th	Soldering and Hardware Implementation in PCB
14th	Report writing and making Video Demonstration

7 References

- https://github.com/NishantSahay7/Nodemcu-to-Google-Sheets
- https://circuitdigest.com/microcontroller-projects/electronic-voting-machine-using-arduino
- https://www.electroniclinic.com/google-spreadsheet-or-google-sheets-with-esp8266-nodemcu-for-data-logging/
- https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library
- https://www.make-it.ca/nodemcu-details-specifications/