CHAPTER 1
INTRODUCTION & OBJECTIVE

Introduction

Elections play a crucial role in democratic systems, ensuring that citizens can choose their leaders in a fair and transparent manner. However, traditional paper-based voting methods are prone to errors, inefficiency, and tampering. To address these challenges, electronic voting machines (EVMs) have been developed to enhance the accuracy, speed, and reliability of the voting process.

This project focuses on designing an EVM using the 8051 microcontroller, a popular embedded system platform. The EVM allows voters to cast their votes electronically using push buttons assigned to candidates. The votes are counted and displayed in real-time on an LCD screen. Additionally, the system uses non-volatile memory (EEPROM) to store vote counts securely, even in case of power failure.

The simplicity and efficiency of the EVM make it suitable for use in small-scale elections, such as student councils, local organizations, or prototype demonstrations for larger systems. By employing microcontroller-based automation, this project ensures a more reliable and user-friendly voting experience.

Objective

The primary objectives of this project are:

1. **Develop a Secure EVM**: Design a reliable electronic voting system using the 8051 microcontroller to minimize errors and enhance data security.
2. **Real-Time Vote Counting**: Implement a system that captures and displays vote counts instantaneously on an LCD.
3. **User Feedback Mechanism**: Provide visual (LED) feedback to indicate successful vote casting.
4. **Cost-Effective Solution**: Create a compact and cost-effective prototype suitable for small-scale elections.

These objectives aim to demonstrate the potential of embedded systems in creating efficient and tamper-proof voting mechanisms for various applications.

CHAPTER 2
Components

▪ **8051 Controller**

▪ **16×2 LCD**

▪ **Pushbuttons**

▪ **Power Supply (5V DC)**

▪ **Connecting Wires**

▪ **Crystal Oscillator (12 Mhz)**

▪ **Potentiometer (10K)**

1. AT89C51 Microcontroller

- **Role**: Acts as the central controller, managing input (from buttons), processing data (vote counts), and output (to the LCD, LEDs, and EEPROM).
- **Key Features**:
  - 4 KB on-chip Flash memory for storing the program.
  - 128 bytes of internal RAM for temporary data storage.
  - 32 input/output (I/O) pins organized in 4 ports (P0–P3).
  - Two 16-bit timers/counters for handling real-time operations.
  - Full-duplex UART for serial communication if needed.
  - Clock speed up to 12 MHz for standard operation.
- **Advantages**: Low-cost, easy-to-program microcontroller with a simple architecture.

---

2. Push Buttons

- **Role**: Capture the votes from users. Each button is assigned to a candidate.
- **Description**:
  - Normally-open momentary switches.
  - When pressed, the button sends a HIGH signal to the AT89C51's I/O pin.
- **Usage**:
  - Each button represents one candidate.
  - A dedicated button can be used to display vote counts or reset the system.
- **Implementation**: Buttons are debounced in software or hardware to prevent multiple detections for a single press.

---

3. LCD Display (16x2)

- **Role**: Displays system messages (e.g., "Vote Now") and the vote counts for each candidate.
- **Description**:

- A 16-column, 2-row alphanumeric display module that interfaces with the microcontroller using either an 8-bit or 4-bit mode.

- **Usage**:
  - Displays instructions to voters.
  - Shows real-time vote counts and final results after polling is complete.
- **Connection**:
  - Requires at least 6 I/O pins from the AT89C51 for data and control signals.

---

4. Power Supply

- **Role**: Supplies power to the microcontroller and other components.
- **Description**:
  - A 5V DC regulated power supply is required.
  - Consists of a step-down transformer, rectifier, voltage regulator (7805 IC), and filter capacitors.
- **Usage**:
  - Powers the AT89C51, LCD, EEPROM, LEDs, and other peripherals.

---

5. Crystal Oscillator (11.0592 MHz)

- **Role**: Provides the clock signal for the AT89C51 to operate.
- **Description**:
  - An 11.0592 MHz crystal oscillator is connected to the XTAL1 and XTAL2 pins of the microcontroller along with capacitors.
- **Usage**:
  - Sets the microcontroller’s clock speed, critical for timing and serial communication.

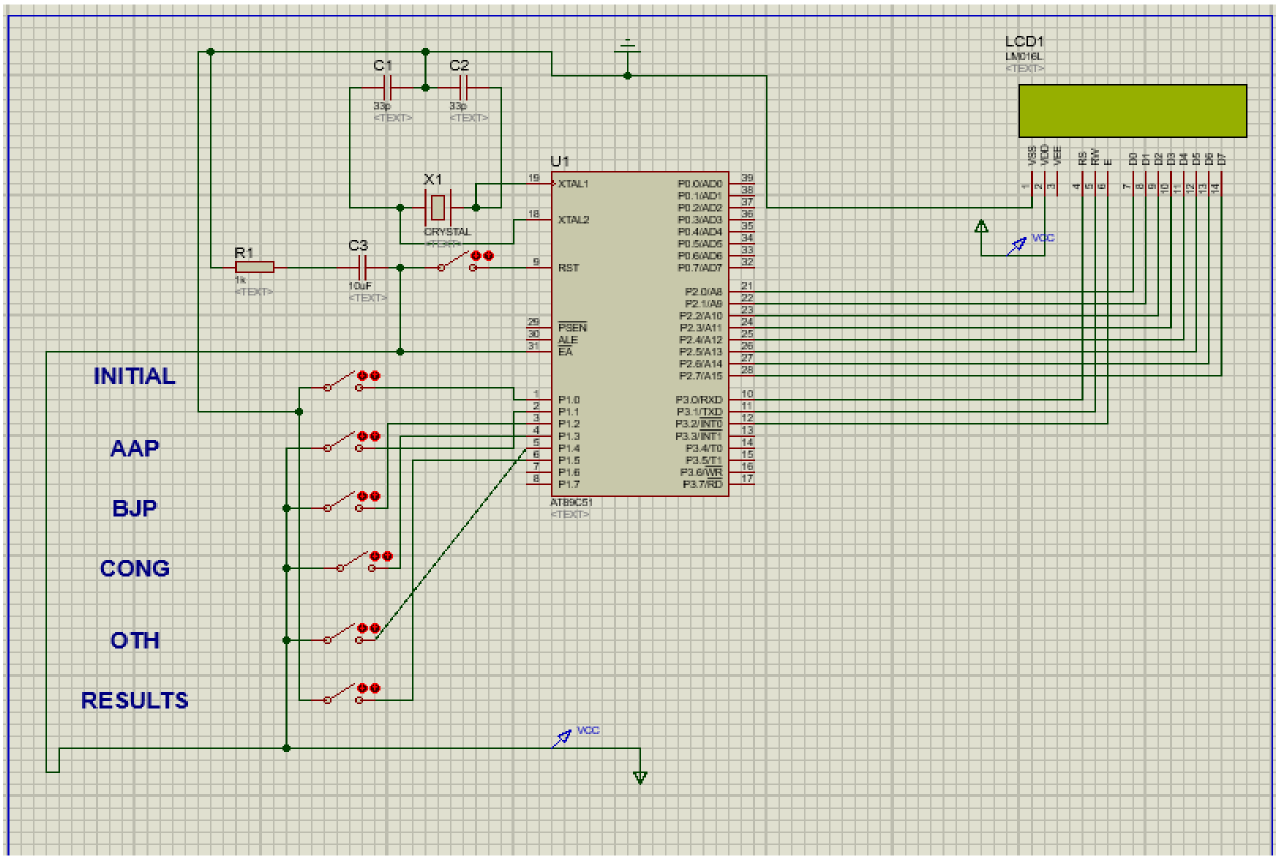
---

6. Resistors and Capacitors

- **Role**: Supporting components for circuit functionality.
- **Description**:
  - Resistors are used for current limiting (e.g., with LEDs and buttons).
  - Capacitors are used for power filtering and clock stabilization.
- **Usage**:
  - Pull-up resistors for buttons to prevent floating states.
  - Decoupling capacitors for stabilizing power to the microcontroller.

CHAPTER 3

Circuit Diagram



Circuit Description:

Here we connected four switches with Port 1 of controller and define accordingly:

- AAP (P1^1)
- BJP (P1^2)
- CONG (P1^3)
- OTH (P1^4)

Initial and result button also connected with Port 1 (0 and 5). Initial button is used to initialize the electronic voting machine and result button will show the final results on LCD after once voting is done.

You can vote multiple times, after just pressing the initial button after one vote.

LCD (Liquid Crystal Display) is used to display the output. Data pins of LCD (7-14) is connected to the Port 2 of 8051 controller and RS, RW & EA pin connected to Port 3 (P3^0, 1, 2) respectively.

Electronic Voting Machine Working:

The voting is begun by squeezing the Initial switch after which the client is incited to vote a ballot. The include of votes is put away in four unique factors. When the client votes in favor of an up-and-comer by squeezing one of the switches, the worth of the relating variable is expanded by one. After this a Thank you message is shown on LCD to recognize the enlistment of client's vote. The message stays on the screen until the following client either presses the Initial button to make another choice or Stop switch is squeezed get the survey results. At the point when the result button is squeezed, the names of the up-and-comers are shown alongside their vote counts. After some deferral, the outcome is shown which could be either statement of the champ up-and-comer or the competitors with a conflict of their number of votes

The system provides immediate feedback to the voter via a buzzer sound or an LED light, confirming the successful registration of the vote. Once the voting process is completed, an administrator can press a specific button to display the results on the LCD. The microcontroller retrieves the stored vote counts from the EEPROM and displays the total votes for each candidate in a sequential manner. This process ensures a secure, efficient, and tamper-resistant voting system, suitable for small-scale elections.

CHAPTER 4
SOFTWARE IMPLIMENTATION

```
// Program to make a voting machine using LCD

#include<reg51.h>
#define msec 50
#define lcd_data_str_pin P2
sbit rs = P3^0;  //Register select (RS) pin
sbit rw = P3^1;  //Read write(RW) pin
sbit en = P3^2;  //Enable(EN) pin
sbit ini_pin = P1^0; // Start voting pin
sbit stop_pin = P1^5; // Stop voting pin

sbit  candidate_1=P1^1;  //Candidate1
sbit  candidate_2=P1^2;  //Candidate2
sbit  candidate_3=P1^3;  //Candidate3
sbit  candidate_4=P1^4;  //Candidate4
int max = 0;
int carry = 0;
int arr[4];

int vote_amt[3],j;
unsigned int vote_1,vote_2,vote_3,vote_4;

void delay(int delay_time)  // Time delay function
{
int j,k;
for(j=0;j<=delay_time;j++)
  for(k=0;k<=1000;k++);
}

void lcd_cmd(unsigned char cmd_addr)  //Function to send command to LCD
{
lcd_data_str_pin = cmd_addr;
en = 1;
rs = 0;
rw = 0;
delay(1);
en = 0;
return;
}

void lcd_data_str(char str[50])  //Function to send string
{
int p;
for (p=0;str[p]!='\0';p++)
{
  lcd_data_str_pin = str[p];
  rw = 0;
  rs = 1;
  en = 1;
  delay(1);
  en = 0;
}
return;
}

void lcd_data_int(unsigned int vote)  //Function to send 0-9 character values
{
char dig_ctrl_var;
int p;
for (j=2;j>=0;j--)
{
  vote_amt[j]=vote%10;
  vote=vote/10;
}
```

```
for (p=0;p<=2;p++)
{
  dig_ctrl_var = vote_amt[p]+48;
  lcd_data_str_pin = dig_ctrl_var;
  rw = 0;
  rs = 1;
  en = 1;
  delay(1);
  en = 0;
}
return;
}

void vote_count()  // Function to count votes
{
while (candidate_1==0 && candidate_2==0 && candidate_3==0 && candidate_4==0);
if (candidate_1==1)
{
  while (candidate_1 == 1);
   {
    vote_1 = vote_1 + 1;
   }
}

if (candidate_2==1)
{
  while (candidate_2 == 1);
   {
    vote_2 = vote_2 + 1;
   }
}

if (candidate_3==1)
{
  while (candidate_3 == 1);
   {
    vote_3 = vote_3 + 1;
   }
}

if (candidate_4==1)
{
  while (candidate_4 == 1);
   {
    vote_4 = vote_4 + 1;
   }
}
}

void lcd_ini()
{
    lcd_cmd(0x38);
delay(msec);
lcd_cmd(0x0E);
delay(msec);
lcd_cmd(0x01);
delay(msec);
lcd_cmd(0x81);
delay(msec);
lcd_data_str("WELCOME!!!");
delay(100);
lcd_cmd(0x01);
delay(msec);
lcd_cmd(0x80);
delay(msec);
lcd_data_str( "PRESS" );
delay(msec);
lcd_cmd(0x14);
delay(msec);
lcd_data_str("BUTTON");
delay(msec);
```

```
delay(msec);
lcd_cmd(0xC0);
delay(msec);
lcd_data_str("TO");
delay(msec);
lcd_cmd(0x14);
delay(msec);
lcd_data_str("VOTE");
delay(100);
lcd_cmd(0x01);
delay(msec);
lcd_cmd(0x80);
delay(msec);
lcd_data_str("AAP");
delay(msec);
lcd_cmd(0x84);
delay(msec);
lcd_data_str("BJP");
delay(msec);
lcd_cmd(0x88);
delay(msec);
lcd_data_str("CON");
delay(msec);
lcd_cmd(0x8C);
delay(msec);
lcd_data_str("OTH");
delay(msec);

vote_count();
lcd_cmd(0x01);
delay(msec);
lcd_cmd(0x85);
delay(msec);
lcd_data_str("THANK");
delay(msec);
lcd_cmd(0x14);
delay(msec);
lcd_data_str("YOU!!");
delay(100);
}

void results()  // Function to show results
{
int i;
carry = 0;
lcd_cmd(0x01);
delay(msec);
lcd_cmd(0x80);
delay(msec);
lcd_data_str("Results");
delay(msec);
lcd_cmd(0x14);
delay(msec);
lcd_data_str("Are");
delay(msec);
lcd_cmd(0x14);
delay(msec);
lcd_data_str("Out");
delay(msec);

  lcd_cmd(0x01);
delay(msec);
lcd_cmd(0x80);
delay(msec);
lcd_data_str("AAP");
delay(msec);
lcd_cmd(0x84);
delay(msec);
lcd_data_str("BJP");
delay(msec);
lcd_cmd(0x88);
```

```
delay(msec);
lcd_data_str("CON");
delay(msec);
lcd_cmd(0x8C);
delay(msec);
lcd_data_str("OTH");
delay(msec);

lcd_cmd(0xC0);
delay(100);
lcd_data_int(vote_1);
delay(msec);

lcd_cmd(0xC4);
delay(msec);
lcd_data_int(vote_2);
delay(msec);

lcd_cmd(0xC8);
delay(msec);
lcd_data_int(vote_3);
delay(msec);

lcd_cmd(0xCC);
delay(msec);
lcd_data_int(vote_4);
delay(300);

arr[0] = vote_1;
arr[1] = vote_2;
arr[2] = vote_3;
arr[3] = vote_4;

for( i=0; i<4; i++)
{
  if(arr[i]>=max)
  max = arr[i];
}

if ( (vote_1 == max) && ( vote_2 != max) && (vote_3 != max)&& (vote_4 != max) )
{
  carry = 1;
  lcd_cmd(0x01);
  delay(msec);
  lcd_cmd(0x82);
  delay(msec);
  lcd_data_str("HURRAY!!!");
  delay(50);
  lcd_cmd(0xC4);
  delay(msec);
  lcd_data_str("AAP");
  delay(msec);
  lcd_cmd(0x14);
  delay(msec);
  lcd_data_str("WINS");
  delay(msec);
}

if ( (vote_2 == max) && ( vote_1 != max) && (vote_3 != max)&& (vote_4 != max) )
{
  carry = 1;
  lcd_cmd(0x01);
  delay(msec);
  lcd_cmd(0x82);
  delay(msec);
  lcd_data_str("HURRAY!!!");
  delay(50);
  lcd_cmd(0xC4);
  delay(msec);
  lcd_data_str("BJP");
  delay(msec);
  lcd_cmd(0x14);
```

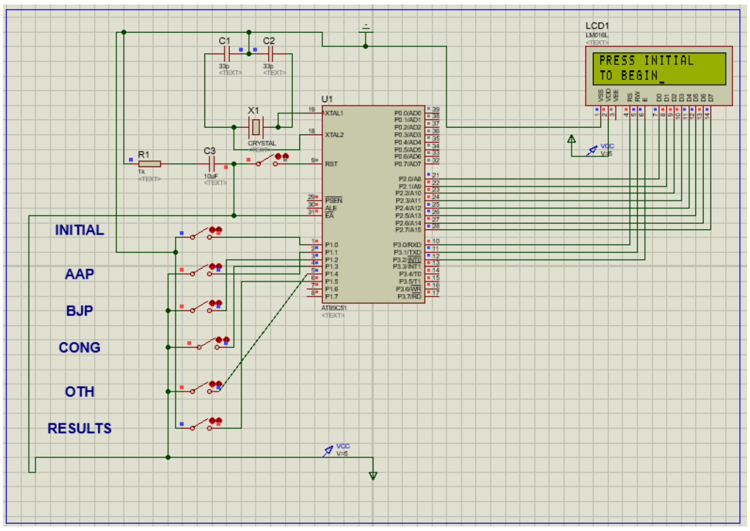
```
    delay(msec);
    lcd_data_str("WINS");
    delay(msec);
  }

  if ( (vote_3 == max) && ( vote_2 != max) && (vote_1 != max)&& (vote_4 != max) )
  {
    carry = 1;
    lcd_cmd(0x01);
    delay(msec);
    lcd_cmd(0x82);
    delay(msec);
    lcd_data_str("HURRAY!!!");
    delay(50);
    lcd_cmd(0xC4);
    delay(msec);
    lcd_data_str("CON");
    delay(msec);
    lcd_cmd(0x14);
    delay(msec);
    lcd_data_str("WINS");
    delay(msec);
  }

  if ( (vote_4 == max) && ( vote_2 != max) && (vote_3 != max)&& (vote_1 != max) )
  {
    carry = 1;
    lcd_cmd(0x01);
    delay(msec);
    lcd_cmd(0x82);
    delay(msec);
    lcd_data_str("HURRAY!!!");
    delay(50);
    lcd_cmd(0xC4);
    delay(msec);
    lcd_data_str("OTH");
    delay(msec);
    lcd_cmd(0x14);
    delay(msec);
    lcd_data_str("WINS");
    delay(msec);
  }

  if (carry==0)
  {
    lcd_cmd(0x01);
    delay(msec);
    lcd_cmd(0x82);
    delay(msec);
    lcd_data_str("CLASH");
    delay(50);
    lcd_cmd(0x14);
    delay(msec);
    lcd_data_str("BETWEEN!!!");
    delay(50);
    if(vote_2 == max)
    {
     lcd_cmd(0xC5);
     lcd_data_str("BJP");
     delay(50);
    }
    if(vote_3 == max)
    {
     lcd_cmd(0xC9);
     lcd_data_str("CON");
     delay(50);
    }
    if(vote_4 == max)
    {
     lcd_cmd(0xCD);
     lcd_data_str("OTH");
     delay(50);
```

```
  }
}
}

void main()
{
ini_pin = stop_pin = 1;
vote_1 = vote_2 = vote_3 = vote_4 = 0;
candidate_1 = candidate_2 = candidate_3 = candidate_4 = 0;
lcd_cmd(0x38);
delay(msec);
lcd_cmd(0x0E);
delay(msec);
lcd_cmd(0x01);
delay(msec);
lcd_cmd(0x80);
delay(msec);
lcd_data_str( "PRESS" );
delay(msec);
lcd_cmd(0x14);
delay(msec);
lcd_data_str("INITIAL");
delay(msec);

delay(msec);
lcd_cmd(0xC0);
delay(msec);
lcd_data_str("TO");
delay(msec);
lcd_cmd(0x14);
delay(msec);
lcd_data_str("BEGIN");
delay(100);
while(1)
{
  while(ini_pin != 0)
  {
   if (stop_pin == 0)
   break;
  }
  if (stop_pin == 0)
  {
  break;
  }
  lcd_ini();
}

while(1)
{
results();
}
}
```

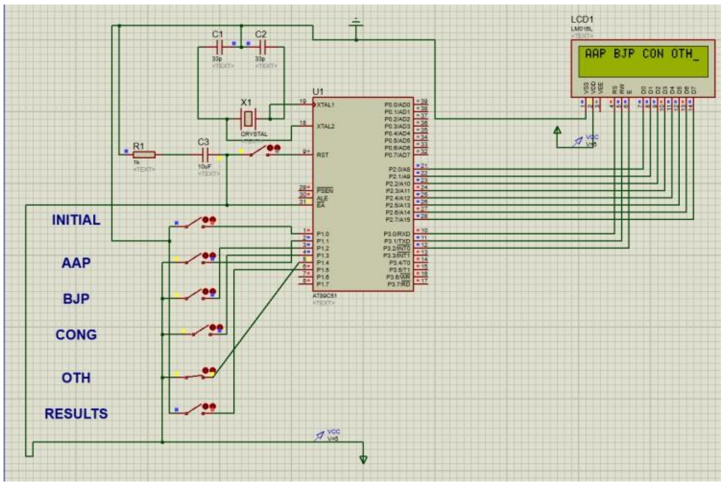
CHAPTER 5

RESULTS

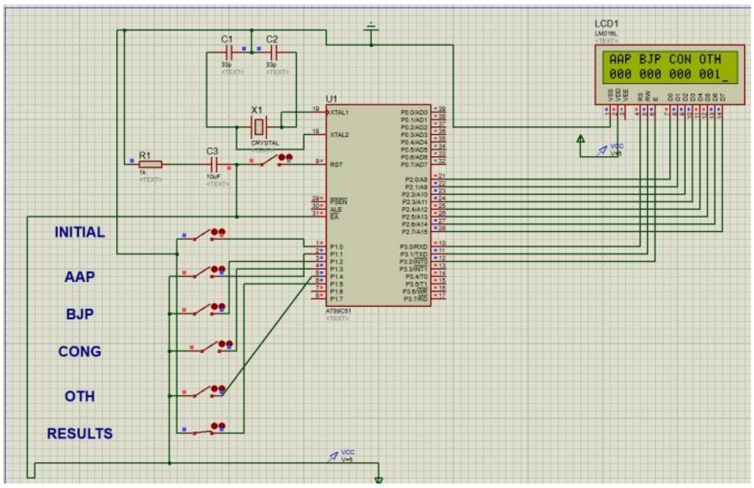
initializing a vote by the govt employee



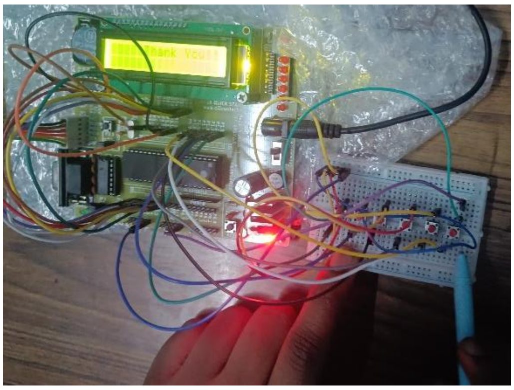
voting a candidate

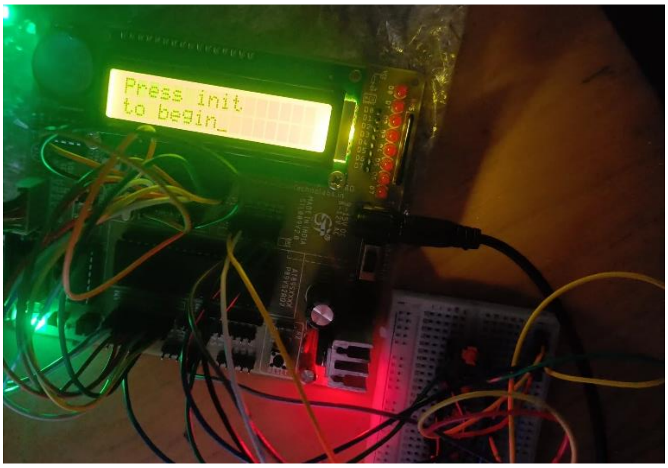


displaying results



HARDWARE RESULTS





PRESSING INTIAL TO BEGIN

VOTING

DISPLAYING RESULTS

CHAPTER 6

APPLICATIONS

The advantages and application of electronic voting machine include the following.

▪ It is inexpensive

▪ It requires less manpower

▪ Time conscious, as less time necessary for polling and counting

▪ Avoids illegal voting

▪ Saves the cost of transportation due to its compact size

▪ Suitable on the voter's part.

▪ This project could be used for the purpose of voting in any necessary place

CHAPTER 7 CONCLUSIONS

The development of an **Electronic Voting Machine (EVM) using the AT89C51 microcontroller** has successfully demonstrated the practical application of embedded systems in automating and enhancing the voting process. The following conclusions can be drawn from this project:

1. **Reliability**: The system provides an accurate and error-free method for recording and counting votes, eliminating the inefficiencies and inaccuracies associated with traditional paper-based voting.
2. **Security**: By integrating EEPROM, the vote data remains persistent even during power outages, ensuring the integrity of the results.
3. **User-Friendly Interface**: With features like push buttons for inputs, an LCD for displaying information, and LEDs/buzzers for feedback, the EVM is intuitive and easy to operate for voters and administrators alike.
4. **Scalability**: The prototype is designed for small-scale elections but can be extended to accommodate larger elections with more candidates or additional security features such as RFID or biometric authentication.
5. **Cost-Effectiveness**: Using commonly available components like the AT89C51 microcontroller, the system is affordable and suitable for educational projects or local elections.

CHAPTER 8

REFERENCES

Tools and Software:

[1] Keil uVision IDE: For writing and debugging embedded C programs for the AT89C51.
[2] Proteus Design Suite: For simulating the circuit before implementation
[3] https://arvingroup.in/electronic-voting-machine-source-code/*)*