ELECTRONIC VOTING MACHINE USING ARDUINO

DE MINI PROJECT

Introduction

An Electronic Voting Machine (EVM) is a device designed to electronically record and count votes securely, accurately, and efficiently. This project demonstrates a simplified EVM using Arduino technology, showcasing the modernization of traditional voting systems through electronics.

Why Arduino?

- Open-source and easy to program: Ideal for prototyping.
- Instant vote counting: Eliminates manual tallying errors.
- Real-time LCD display: Provides live results.
- Standalone system: No Wi-Fi/Bluetooth connectivity ensures protection against remote hacking.
- Tamper-resistant: Votes are stored locally, unlike online voting systems.

Components Required

The EVM system utilizes the following hardware components:

- Arduino Uno Board: The microcontroller processes inputs and controls outputs. It is programmed using an IDE.
- LCD 16x2 I2C: Displays voting options, results, and prompts using the I2C protocol.
- Push Buttons: Used for casting votes.
- Resistors: Current-limiting and pull-down configurations.
- Breadboard: For circuit prototyping.
- Jumper Wires: Connect various components.
- LEDs: Indicate successful voting and display results.
- Piezo Buzzer: Provides audio feedback during voting.
- Keypad 4x4 Matrix: Enables user authentication via roll number input.

Circuit Diagram and Connections

The circuit diagram illustrates how the components are interconnected:

Arduino Uno Connections

1. Power and Ground:
    - VIN connected to the 5V power supply.
    - GND connected to the ground rail.
2. Digital Pins:
    - D2–D5 connected to keypad columns (C1–C4).
    - D6–D9 connected to keypad rows (R1–R4).
    - D10 connected to the buzzer.
    - D11 connected to the green LED (via resistor R1).
    - D12 connected to the red LED.
3. Analog Pins:
    - A0–A4 connected to push buttons (F1–F5).
    - A5 (SDA) and A4 (SCL) connected to the LCD module for I2C communication.

Additional Connections:

- Push Buttons (F1–F5):
    - One terminal connected to analog pins A0–A4 respectively; other terminal connected to ground.
- Keypad Connections:
    - Rows (R1–R4) connected to digital pins D6–D9.
    - Columns (C1–C4) connected to digital pins D2–D5.
- LCD Connections:
    - VCC connected to 5V power supply; GND connected to ground; SDA and SCL connected to A5 and A4 respectively.

Working of the EVM

The EVM operates in four key stages:

1. User Authentication

The user enters their roll number via the keypad. This number is validated against a pre-defined database. If valid and unused, the user is authenticated.

2. Voting

Authenticated users are presented with a voting screen on the LCD. They cast their votes by pressing one of four buttons corresponding to candidates (A, B, C, or OTH for "Other").

3. Vote Recording

Votes are recorded securely, and users are marked as having voted to prevent duplicate entries.

4. Results Display

Pressing the result button displays election results on the LCD, indicating the candidate with the highest votes. After a delay, the system resets for new users.

Hardware Components Overview

1. Arduino UNO: Acts as the brain of the EVM, controlling all components including LCD, keypad, buttons, LEDs, and buzzer.
2. LCD (16x2): Displays prompts, voting options, results, and system messages via I2C communication.
3. Keypad (4x4 Matrix): Facilitates roll number input for authentication.
4. Push Buttons (SW1–SW5):
    - SW1–SW4 represent candidates A, B, C, and OTH respectively.
    - SW5 displays election results when pressed.
5. Green LED (D12): Indicates successful voting.
6. Red LED (D13): Indicates result display mode.
7. Buzzer (D12): Provides audio feedback during button presses.

Software Logic

The software is programmed using Arduino IDE with libraries like Adafruit_LiquidCrystal.h for LCD control and Keypad.h for handling keypad input.

Key Functions:

1. Setup Phase:
    - Configures button pins as inputs with internal pull-up resistors (`pinMode(sw1, INPUT); digitalWrite(sw1, HIGH);`).
    - Initializes LCD (`lcd_1.begin(16, 2);`) and displays login prompt (`promptForLogin();`).
2. Loop Phase:
    - Continuously checks for result button press (`SW5`). If pressed, calls `showResults()` function.
    - Handles user login (`handleLogin()`) if not authenticated.
    - Processes voting (`handleVoting()`) after authentication.
3. Authentication Logic:
    - Reads keypad input (`handleLogin()`), converts roll number into an integer, and validates it against predefined roll numbers (`authenticateUser()`).
4. Voting Logic:
    - Records votes (`recordVote()`) by incrementing counters for respective candidates and marks users as having voted.
5. Result Display Logic:
    - Determines winner based on vote counts (`showResults()`) and resets system for new users after displaying results.

Need & Advantages

The project addresses several needs in electronic voting systems:

- Secure user authentication via roll number input on a keypad.
- Clear candidate selection with dedicated buttons for each option including "OTH" for abstention votes.
- Real-time feedback through I2C LCD display and buzzer alerts.
- Transparent result indication using LEDs.

Advantages include faster vote counting without manual errors, tamper-resistant local storage of votes, and elimination of risks associated with online systems like hacking or data breaches.

Conclusion

This project successfully demonstrates a simplified yet effective electronic voting machine using Arduino technology. By integrating hardware components like LCDs, keypads, LEDs, push buttons, and buzzers with robust software logic, it ensures secure authentication, accurate vote recording, real-time result display, and tamper-resistant operation – making it an ideal prototype for modernizing traditional electoral systems.

WRITTEN BY:

Tejas Shinde 23EEB0A043

Taksheel Dupare 23EEB0A59

Ajwad Habib Anwer Sadath 23EEB0F04