**BVC College of Engineering Rajamundry**

**ARM Module**

**Batch: EMBBVC 02**

Training Dates: 09/06/2025 to 21/06/2025

**Digital Voting Machine Using LPC1768 with LCD and Keypad Interface**

Gatti Kanaka Mahalakshmi CL20250304011013610

Kollepara Padmavathi CL20250304011014216

Malla Jahnavi Sri Sai Tulasi CL20250304011014317

**Trainer Name  
Vyasaraj T**

**Contents**

* INTRODUCTION
* OBJECTIVES
* HARDWARE AND SOFTWARE REQUIREMENTS
* METHODOLOGY
* Description of The Project
* SOURCE CODE
* CODE EXPLANATION
* RESULTS
* CONCLUSION

**INTRODUCTION**

A voting machine is an electronic device used to record, store, and count votes during an election. It replaces traditional paper-based voting systems, making the process faster, more accurate, and less prone to human error. These machines ensure a secure, tamper-resistant, and transparent voting environment. In this project, we implement a basic electronic voting machine using the LPC1768 microcontroller. The system allows users to cast their votes for three candidates by pressing buttons, and displays the voting results on an LCD screen. It is an excellent example of how embedded systems can be used to develop real-world applications with user input handling, data

storage, and display output.

**OBJECTIVES**

* Design and develop a simple electronic voting system using the LPC1768 microcontroller
* To implement an **LCD interface** for real-time display of voting instructions, options, and results.
* To use a **matrix keypad** for capturing voter input and votes securely.
* To accurately **count and store votes** for multiple parties along with NOTA (None of the Above) option.
* To integrate a **buzzer system** for audible confirmation of successful vote casting.
* To provide a **start, stop, and result display** functionality for controlled voting sessions.

**HARDWARE AND SOFTWARE REQUIREMENTS**

• Microcontroller: LPC1768 Development Board

• Input Device: 4x4 Keypad

• Output Devices: 16x2 LCD Display, Buzzer

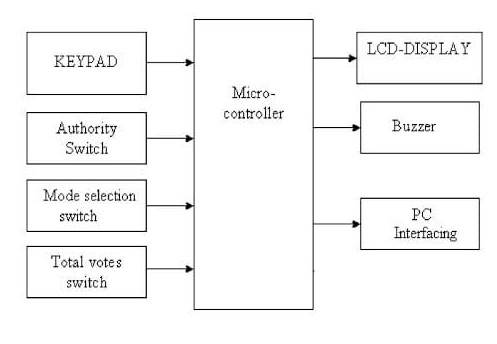
• Power Supply: 5V DC Power Supply

• Connectors and Wires: Jumper Wires, Header Connectors

• Cables: USB Cable

• Programming Tools: PC/Laptop, Keil µVision4 followed by flash magic.

**METHODOLOGY**

****

**Block Diagram of Digital Voting Machine Using LPC1768**

**Description of The Project**

The Digital Voting Machine (DVM) developed using the LPC1768 microcontroller is designed to simulate a secure and user-friendly electronic voting system. It integrates a 4x4 matrix keypad to allow users to cast their votes and a 16x2 LCD display to guide the voting process and show relevant messages such as party names, voting instructions, and results. The LPC1768, based on the ARM Cortex-M3 architecture, handles all the logic, including vote counting, invalid vote detection, and displaying totals. The system starts in an idle state and begins accepting votes once initiated. After all votes are cast, results can be displayed at the press of a specific key. This project showcases the efficient use of embedded systems for real-time voting applications with minimal hardware.

**SOURCE CODE**

#include "lpc17xx.h"

#include "lcd.h"

#include "keypad.h"

#include <stdio.h>

#define NUM\_PARTIES 3 // Number of valid parties

#define NOTA '5' // Key for invalid votes

#define VOTING\_START\_KEY '1' // Key to start voting

#define VOTING\_STOP\_KEY '6' // Key to stop voting

#define RESULTS\_KEY '7' // Key to display results

uint8\_t vote\_count[NUM\_PARTIES] = {0}; // Vote counters for each party

uint8\_t invalid\_votes = 0; // Counter for invalid votes

uint8\_t voting\_active = 0; // Voting status

uint8\_t voting\_stopped = 0; // Voting stopped status

#define BUZZER\_PIN (1 << 27) // Assuming buzzer is connected to P1.27

void delay\_ms(uint32\_t ms);

void buzzer\_sound(void);

int main(void) {

char key;

char message[32];

// Initialize peripherals

lcd\_init();

keypad\_init();

// Initialize GPIO for buzzer

LPC\_GPIO1->FIODIR |= BUZZER\_PIN; // Set P2.10 as output

LPC\_GPIO1->FIOCLR = BUZZER\_PIN; // Turn off buzzer initially

while (1) {

if (!voting\_active && !voting\_stopped)

{

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Digital Voting");

lcd\_cmd\_write(0xC0); // Clear LCD

lcd\_str\_write("Press 1 to Start");

}

key = keypad\_getkey(); // Wait for user input

if (key == VOTING\_START\_KEY && !voting\_active && !voting\_stopped)

{

voting\_active = 1;

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Party 1:A");

lcd\_cmd\_write(0xC0); // Move to 2nd line

lcd\_str\_write("Party 2:B Party 3:C");

delay\_ms(300);

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Cast Your Vote");

}else if (key == VOTING\_STOP\_KEY &&

voting\_active) {

voting\_active = 0;

voting\_stopped = 1;

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Voting Stopped");

} else if (voting\_active) {

if (key >= '2' && key <= '4')

{ // Votes for Party 1 to 3

uint8\_t party = key - '2'; // Convert key to index (0 to 2)

vote\_count[party]++;

buzzer\_sound(); // Buzzer sound for each vote

lcd\_cmd\_write(0x01); // Clear LCD

sprintf(message, "Voted: Party %c", 'A' + party);

lcd\_str\_write(message);

lcd\_cmd\_write(0xC0);

lcd\_str\_write("Thanks You");

delay\_ms(500); // Briefly show the message

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Cast Your Vote");

}

else if (key ==NOTA)

{ // Nota vote

invalid\_votes++;

buzzer\_sound(); // Buzzer sound for Nota vote

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Nota");

delay\_ms(500); // Briefly show the message

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Cast Your Vote");

}

} else if (key == RESULTS\_KEY && voting\_stopped)

{

lcd\_cmd\_write(0x01); // Clear LCD

sprintf(message, "P1: %d P2: %d", vote\_count[0],

vote\_count[1]);

lcd\_str\_write(message);

lcd\_cmd\_write(0xC0); // Move to 2nd line

sprintf(message, "P3: %d Inv: %d", vote\_count[2],

invalid\_votes);

lcd\_str\_write(message);

delay\_ms(3000); // Display results for 3 seconds

lcd\_cmd\_write(0x01); // Clear LCD

lcd\_str\_write("Results Shown");

delay\_ms(2000); // Allow time to view the message

}

}

}

void delay\_ms(uint32\_t ms) {

uint32\_t i;

for (i = 0; i < ms \* 4000; i++); // Approximate delay loop

}

void buzzer\_sound(void) {

LPC\_GPIO1->FIOSET = BUZZER\_PIN; // Turn on buzzer

delay\_ms(100); // Buzzer ON duration

LPC\_GPIO1->FIOCLR = BUZZER\_PIN; // Turn off buzzer

}

**REGISTERS USED**

**GPIO Registers1. FIODIR (Fast GPIO Direction Register)**LPC\_GPIO1->FIODIR |= BUZZER\_PIN; // Set P2.10 as output**2. FIOSET (Fast GPIO Set Register)**LPC\_GPIO1->FIOSET = BUZZER\_PIN; // Turn on buzzer**3. FIOCLR (Fast GPIO Clear Register:**

LPC\_GPIO1->FIOCLR = BUZZER\_PIN; // Turn off buzzer

**Pins Used**

1. Buzzer Pin

2. LCD Pins3. Keypad Pins

**CODE EXPLANATION**

**Header Files Used:**

* lpc17xx.h: Contains definitions for LPC1768 microcontroller registers.
* lcd.h: Handles LCD display functions.
* keypad.h: Manages keypad scanning and key input.
* stdio.h: Used for string formatting via sprintf().

**Macro Definitions:**

* NUM\_PARTIES: Defines number of valid political parties (3).
* Key mappings:
* '1' – Start Voting
* '2' to '4' – Cast Vote for Party A, B, or C
* '5' – Nota vote
* '7' – Display Result
* '6' – Stop Voting
* BUZZER\_PIN: Defines the buzzer pin (P1.27).

**Global Variables:**

* vote\_count[]: Stores individual vote counts for each party.
* invalid\_votes: Stores count of invalid votes.
* voting\_active and voting\_stopped: Flags to manage voting states.

**Peripheral Initialization:**

* LCD and Keypad initialized using their respective functions.
* Buzzer pin is configured as output and initially turned off.

**User Interface Logic:**

* When the system starts, LCD displays “Digital Voting” and prompts user to press '1' to start.
* Pressing '1' activates the voting phase and shows available parties (A, B, C).

**Voting Logic:**

* Pressing '2', '3', or '4' registers a vote for Party A, B, or C.
* Corresponding party's vote count is incremented.
* A buzzer sound is played for feedback.
* A “Vote Done” and “Thank You” message is displayed on LCD.

**Invalid Voting:**

* Pressing '5' is considered an Nota vote.
* Increments invalid vote counter.
* Shows “Nota” on LCD with buzzer feedback.

**Stopping Voting:**

* Pressing '6' ends the voting process.
* System shows “Voting Stopped” and disables further voting.

**Displaying Results:**

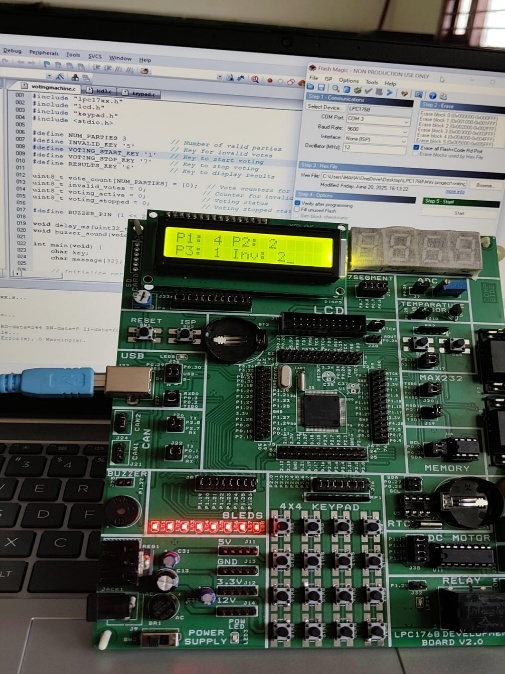
* Pressing '7' after voting is stopped displays the result:
* Vote counts for Party A, B, and C.
* Total invalid votes.

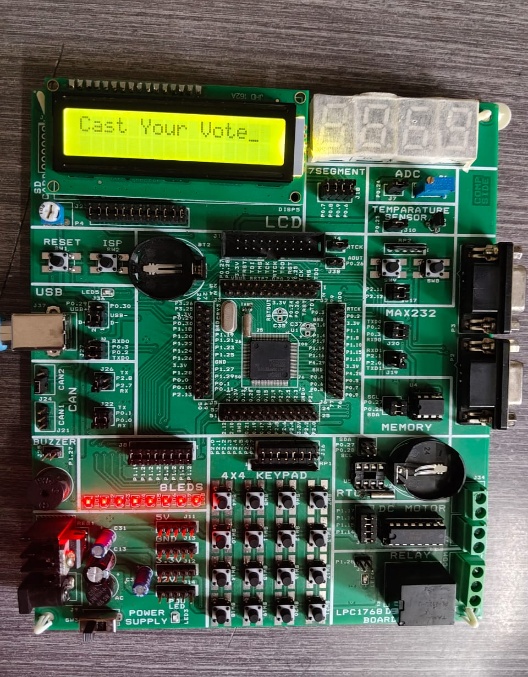
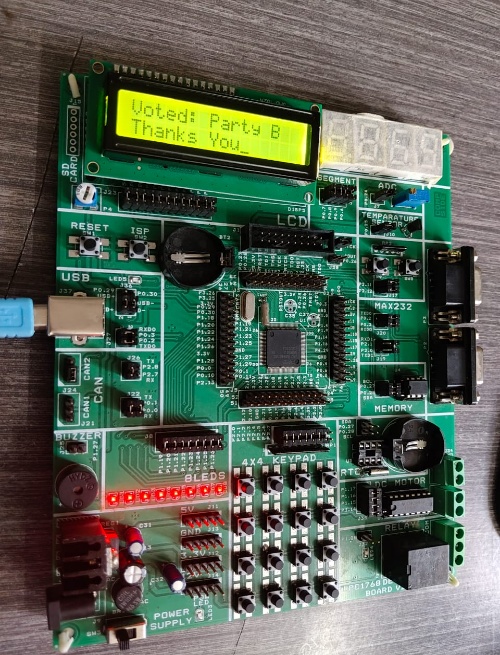
**Helper Functions:**

* delay\_ms(): Software-based delay function used for timing LCD/buzzer.
* buzzer\_sound(): Turns buzzer ON briefly to acknowledge input.
* 

**RESULT**

The Digital Voting Machine accurately recorded and displayed votes using the LPC1768, LCD, and keypad. It successfully processed valid inputs and showed correct results after voting ended.





**CONCLUSION**

The Digital Voting Machine using LPC1768 with LCD and keypad successfully demonstrates a simple and efficient electronic voting system. It ensures accurate vote recording, user-friendly interaction, and reliable result display. The project highlights the practical use of embedded systems in real time applications. It also improves understanding of microcontroller interfacing and logic design. Overall, the system is cost-effective and suitable for small-scale voting scenarios.