#include <avr/io.h>

#include <avr/interrupt.h>

#include <string.h>

#include <stdlib.h>

#define *F\_CPU* 1000000

#include <util/delay.h>

#define E PD5

#define RS PD6

volatile *uint16\_t* timer\_ms = 0;

volatile *uint8\_t* timer\_flag = 0;

void send\_a\_command(unsigned char command);

void send\_a\_character(unsigned char character);

void send\_a\_string(char \*string\_of\_characters);

void lcd\_init();

void menu();

void listOfCandidates();

void votingFunction();

void showResults();

void winnerAnnouncement();

void infoMenu();

unsigned char getKey();

void waitForKeyRelease();

// Timer functions

void timer\_init();

void delay\_ms(*uint16\_t* ms);

void start\_timer(*uint16\_t* ms);

*uint8\_t* timer\_expired();

char candidatesName[6][10] = {"Akshay", "Vikas", "Swaroop", "Suhas", "Sena", "Janaki"};

int candidateIDs[6] = {16, 24, 58, 62, 37, 49};

int votes[6] = {0, 0, 0, 0, 0, 0};

int totalVotes = 0;

unsigned char keypad[4][3] = {

{'1', '4', '7','\*'},

{'2', '5', '8','0'},

{'3', '6', '9','#'},

};

ISR(TIMER0\_OVF\_vect)

{

if(timer\_ms > 0)

{

timer\_ms--;

if(timer\_ms == 0)

{

timer\_flag = 1;

}

}

TCNT0 = 240;

}

int main(void)

{

DDRA = 0xFF;

DDRD = 0xFF;

DDRB = 0x0F;

PORTB = 0xF0;

timer\_init();

sei();

lcd\_init();

send\_a\_command(0x01);

send\_a\_string(" WELCOME TO ");

send\_a\_command(0xC0);

send\_a\_string("VOTING SYSTEM ");

delay\_ms(3000);

menu();

return 0;

}

void timer\_init()

{

TCCR0 = (1<<CS01) | (1<<CS00); // Prescaler 64

TCNT0 = 240; // Initial value for ~1ms overflow

TIMSK |= (1<<TOIE0); // Enable Timer0 overflow interrupt

}

void delay\_ms(*uint16\_t* ms)

{

start\_timer(ms);

while(!timer\_expired())

{

// Can do other tasks here if needed

}

}

void start\_timer(*uint16\_t* ms)

{

cli(); // Disable interrupts temporarily

timer\_ms = ms;

timer\_flag = 0;

sei(); // Re-enable interrupts

}

*uint8\_t* timer\_expired()

{

return timer\_flag;

}

void lcd\_init()

{

delay\_ms(50);

send\_a\_command(0x38); // Function set: 8-bit, 2 lines, 5x7 font

delay\_ms(5);

send\_a\_command(0x38); // Function set (repeat for reliability)

delay\_ms(1);

send\_a\_command(0x38); // Function set (repeat for reliability)

delay\_ms(1);

send\_a\_command(0x0C); // Display ON, cursor OFF, blink OFF

delay\_ms(1);

send\_a\_command(0x06); // Entry mode: increment cursor, no shift

delay\_ms(1);

send\_a\_command(0x01); // Clear display

delay\_ms(2);

}

unsigned char getKey()

{

unsigned char key = 0;

for(int row = 0; row < 4; row++)

{

// Set all rows high, then set current row low

PORTB = (PORTB & 0xF0) | (0x0F & ~(1 << row));

delay\_ms(5);

// Check each column

for(int col = 0; col < 4; col++)

{

if(!(PINB & (1 << (col + 4))))

{

key = keypad[row][col];

waitForKeyRelease();

return key;

}

}

}

// Reset all rows to high

PORTB = (PORTB & 0xF0) | 0x0F;

return 0;

}

void waitForKeyRelease()

{

while(1)

{

int keyPressed = 0;

for(int row = 0; row < 4; row++)

{

PORTB = (PORTB & 0xF0) | (0x0F & ~(1 << row));

delay\_ms(2);

if((PINB & 0xF0) != 0xF0) // Check if any column is pressed

{

keyPressed = 1;

break;

}

}

PORTB = (PORTB & 0xF0) | 0x0F; // Reset rows

if(!keyPressed) break;

}

delay\_ms(100); // Additional debounce delay

}

void menu()

{

unsigned char key;

while(1)

{

send\_a\_command(0x01);

send\_a\_string("CHOOSE ACTION: ");

send\_a\_command(0xC0);

send\_a\_string("1.LIST 2.VOTE ");

delay\_ms(2000);

send\_a\_command(0x01);

send\_a\_string("3.RESULTS ");

send\_a\_command(0xC0);

send\_a\_string("4.INFO ");

delay\_ms(2000);

send\_a\_command(0x01);

send\_a\_string("Press 1-4: ");

send\_a\_command(0xC0);

send\_a\_string(" ");

while(1)

{

key = getKey();

if(key >= '1' && key <= '4')

{

send\_a\_command(0xC0);

send\_a\_string("Selected: ");

send\_a\_character(key);

delay\_ms(500);

break;

}

}

switch(key)

{

case '1': listOfCandidates(); break;

case '2': votingFunction(); break;

case '3': showResults(); break;

case '4': infoMenu(); break;

}

}

}

void listOfCandidates()

{

unsigned char key;

for(int i = 0; i < 6; i++)

{

send\_a\_command(0x01);

send\_a\_string(candidatesName[i]);

send\_a\_command(0xC0);

send\_a\_string("ID: ");

char idStr[10];

*itoa*(candidateIDs[i], idStr, 10);

send\_a\_string(idStr);

delay\_ms(2500);

}

send\_a\_command(0x01);

send\_a\_string("1.VOTE 2.MENU ");

send\_a\_command(0xC0);

send\_a\_string("Choose: ");

while(1)

{

key = getKey();

if(key == '1') { votingFunction(); return; }

else if(key == '2') return;

}

}

void votingFunction()

{

unsigned char key;

char input[3] = {0};

int inputIndex = 0;

int voteID = 0;

send\_a\_command(0x01);

send\_a\_string("Enter ID (2-digit)");

send\_a\_command(0xC0);

send\_a\_string("ID: ");

while(1)

{

key = getKey();

if(key >= '0' && key <= '9' && inputIndex < 2)

{

input[inputIndex++] = key;

send\_a\_character(key);

if(inputIndex == 2)

{

voteID = (input[0] - '0') \* 10 + (input[1] - '0');

int candidateIndex = -1;

// Find candidate by ID

for(int i = 0; i < 6; i++)

{

if(candidateIDs[i] == voteID)

{

candidateIndex = i;

break;

}

}

if(candidateIndex != -1)

{

votes[candidateIndex]++;

totalVotes++;

send\_a\_command(0x01);

send\_a\_string("VOTE RECORDED! ");

send\_a\_command(0xC0);

send\_a\_string("For: ");

send\_a\_string(candidatesName[candidateIndex]);

delay\_ms(3000);

return;

}

else

{

send\_a\_command(0x01);

send\_a\_string("INVALID ID! ");

send\_a\_command(0xC0);

send\_a\_string("Try again... ");

delay\_ms(2000);

inputIndex = 0;

input[0] = input[1] = 0;

send\_a\_command(0x01);

send\_a\_string("Enter ID (2-digit)");

send\_a\_command(0xC0);

send\_a\_string("ID: ");

}

}

}

else if(key == '\*')

{

// Clear input

inputIndex = 0;

input[0] = input[1] = 0;

send\_a\_command(0xC0);

send\_a\_string("ID: ");

send\_a\_command(0xC4);

}

else if(key == '#')

{

// Exit voting

return;

}

}

}

void showResults()

{

unsigned char key;

char voteStr[10];

send\_a\_command(0x01);

send\_a\_string("TOTAL VOTES: ");

send\_a\_command(0xC0);

*sprintf*(voteStr, "%d", totalVotes);

send\_a\_string(voteStr);

send\_a\_string(" votes");

delay\_ms(2500);

for(int i = 0; i < 6; i++)

{

send\_a\_command(0x01);

send\_a\_string(candidatesName[i]);

send\_a\_command(0xC0);

send\_a\_string("Votes: ");

*sprintf*(voteStr, "%d", votes[i]);

send\_a\_string(voteStr);

delay\_ms(2500);

}

send\_a\_command(0x01);

send\_a\_string("1.WINNER 2.MENU ");

send\_a\_command(0xC0);

send\_a\_string("Choose: ");

while(1)

{

key = getKey();

if(key == '1') { winnerAnnouncement(); return; }

else if(key == '2') return;

}

}

void winnerAnnouncement()

{

int maxVotes = 0;

int winnerIndex = 0;

char voteStr[10];

// Find candidate with maximum votes

for(int i = 0; i < 6; i++)

{

if(votes[i] > maxVotes)

{

maxVotes = votes[i];

winnerIndex = i;

}

}

// Check for tie

int tieCount = 0;

for(int i = 0; i < 6; i++)

{

if(votes[i] == maxVotes) tieCount++;

}

if(tieCount > 1)

{

send\_a\_command(0x01);

send\_a\_string("TIE DETECTED! ");

send\_a\_command(0xC0);

send\_a\_string("Multiple winners");

delay\_ms(3000);

}

else if(maxVotes == 0)

{

send\_a\_command(0x01);

send\_a\_string("NO VOTES CAST! ");

send\_a\_command(0xC0);

send\_a\_string("No winner ");

delay\_ms(3000);

}

else

{

send\_a\_command(0x01);

send\_a\_string("THE WINNER IS: ");

send\_a\_command(0xC0);

send\_a\_string(candidatesName[winnerIndex]);

delay\_ms(3000);

send\_a\_command(0x01);

send\_a\_string(candidatesName[winnerIndex]);

send\_a\_command(0xC0);

send\_a\_string("Votes: ");

*sprintf*(voteStr, "%d", maxVotes);

send\_a\_string(voteStr);

delay\_ms(3000);

}

send\_a\_command(0x01);

send\_a\_string("Press any key ");

send\_a\_command(0xC0);

send\_a\_string("to continue... ");

while(!getKey());

}

void infoMenu()

{

send\_a\_command(0x01);

send\_a\_string("VOTING SYSTEM ");

send\_a\_command(0xC0);

send\_a\_string("Version 1.0 ");

delay\_ms(2500);

send\_a\_command(0x01);

send\_a\_string("Developed by: ");

send\_a\_command(0xC0);

send\_a\_string("Team 5 ");

delay\_ms(2500);

send\_a\_command(0x01);

send\_a\_string("Press any key ");

send\_a\_command(0xC0);

send\_a\_string("to exit ");

while(!getKey());

}

void send\_a\_command(unsigned char command)

{

PORTA = command;

PORTD &= ~(1<<RS); // RS = 0 for command

// RW is connected to ground (always write mode)

PORTD |= (1<<E); // Enable pulse high

delay\_ms(2);

PORTD &= ~(1<<E); // Enable pulse low

delay\_ms(2);

}

void send\_a\_character(unsigned char character)

{

PORTA = character;

PORTD |= (1<<RS); // RS = 1 for data

// RW is connected to ground (always write mode)

PORTD |= (1<<E); // Enable pulse high

delay\_ms(2);

PORTD &= ~(1<<E); // Enable pulse low

delay\_ms(2);

}

void send\_a\_string(char \*string\_of\_characters)

{

while(\*string\_of\_characters > 0)

{

send\_a\_character(\*string\_of\_characters++);

}

}