

**NUST COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING**

MICROCONTROLLERS AND MICROPROCESSOR

PROJECT REPORT

***Submitted by:***

*NC Warda Ahmed*

*ASC Tooba Waqar*

**DE-CE-37**

**Syndicate B**

**Project Description:**

The project that was uptaken to perform was a portable voting machine, the machine is capable of providing a portable solution for citizens residing in remote areas who want to vote but are deprived due to remoteness and distance from the respective voting centers.

The project was divided into two modules, which are as followed:

1. To create a voting machine that is capable of performing voting and computing the result
2. Adding modules of WiFi and GPS for tracking and sending live results to the respective authorities.

Due to the unavailability of resources and funding i.e. each module costing from a range of PKR 2500 to PKR 8000, we aimed for the first module.

**Components:**

* PIC18F452 Microcontroller
* LM016L LCD
* Resistors
* LEDs
* Crystal
* Buzzer
* Buttons
* Capacitors

**Working:**

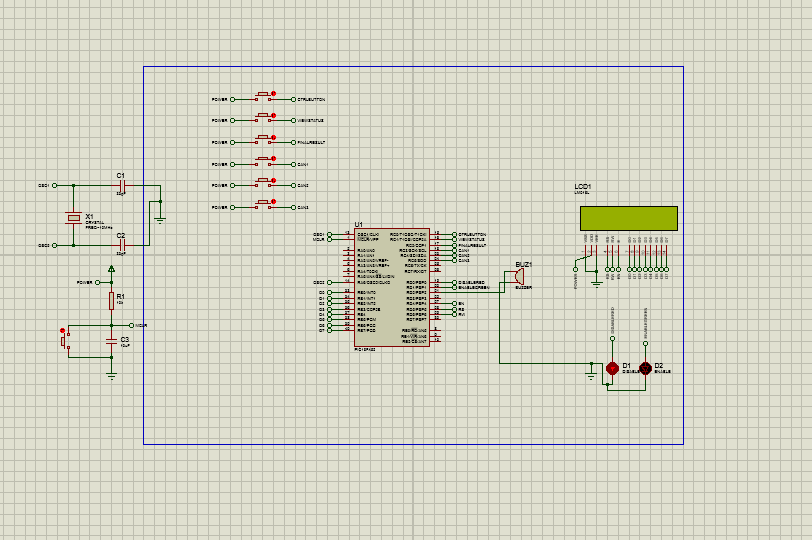
In normal mode -> Voting is disabled (Red LED is ON, Green LED is OFF) Press Control Button to enable (GREEN on, RED off)

ENABLE mode -> Voting by pressing CAN1, CAN2 or CAN3 As soon as CAN1 or CAN2 or CAN3 is pressed, vote is cast and machine goes back into DISABLE mode (hence one can not vote twice)

ENABLE mode -> VIEW STATUS button -> displays the number of votes for each candidate

ENABLE mode -> FINAL RESULT button -> displays the winner of the voting & resets all votes to 0

**Proteus Schematic:**



**Code:**

|  |  |
| --- | --- |
|  | #include <p18f452.h>  #define RW PORTDbits.RD6 |
|  | #define RS PORTDbits.RD5 |
|  | #define E PORTDbits.RD4 |
|  | #define RED PORTDbits.RD0 |
|  | #define GREEN PORTDbits.RD1 |
|  | #define BUZZER PORTDbits.RD2 |
|  | #define CTRLBUTTON PORTCbits.RC0 |
|  | #define VIEWSTATUS PORTCbits.RC1 |
|  | #define RESULT PORTCbits.RC2 |
|  | #define CAN1 PORTCbits.RC3 |
|  | #define CAN2 PORTCbits.RC4 |
|  | #define CAN3 PORTCbits.RC5 |
|  | #define lcd PORTB |
|  |  |
|  | //configs for CONFIG1H, CONFIG2L, CONFIG2H, CONFIG4L |
|  | #pragma config WDT=OFF, DEBUG =OFF, LVP = OFF, OSC = HS |
|  |  |
|  | unsigned char msg[]="Welcome to"; |
|  | unsigned char msg2[] = "Voting Machine"; |
|  | unsigned char voting[] = "C\_1 C\_2 C\_3"; |
|  | unsigned char xxx[] = "\* \* \* "; |
|  | unsigned char DV[] = "DV"; |
|  | unsigned char EV[] = "E"; |
|  | unsigned char cast[] = "Casting vote to"; |
|  | unsigned char cast1[] = "Candidate 1..."; |
|  | unsigned char cast2[] = "Candidate 2..."; |
|  | unsigned char cast3[] = "Candidate 3..."; |
|  | unsigned char winner1[] = "Winner is C\_1"; |
|  | unsigned char winner2[] = "Winner is C\_2"; |
|  | unsigned char winner3[] = "Winner is C\_3"; |
|  | unsigned char congrats[] = "Congratulations!"; |
|  | unsigned char wrong[] = "Something is"; |
|  | unsigned char wrong2[] = "wrong!!!"; |
|  | far unsigned int can1 = 0; |
|  | far unsigned int can2 = 0; |
|  | far unsigned int can3 = 0; |
|  | unsigned char votes1[5]; |
|  | unsigned char votes2[5]; |
|  | unsigned char votes3[5]; |
|  |  |
|  |  |
|  | void displayWelcome(void); |
|  | void enable(void); |
|  | void viewstatus(void); |
|  | void disable(void); |
|  |  |
|  | void delay1(unsigned int i); |
|  | void LCD\_CMD(unsigned char i); |
|  | void LCD\_DATA(unsigned char i ); |
|  | void display(unsigned char \* ); |
|  | void Sendchar(unsigned char ); |
|  |  |
|  | void main() |
|  | { |
|  | TRISDbits.TRISD0=0; //red led output |
|  | TRISDbits.TRISD1=0; // green led output |
|  | TRISDbits.TRISD2=0; // buzzer |
|  | TRISC = 0xFF; // buttons |
|  | TRISDbits.TRISD4=0; // E of LCD |
|  | TRISDbits.TRISD5=0; // RS of LCD |
|  | TRISDbits.TRISD6=0; // RW of LCD |
|  | TRISB=0X00; // LCD data pins |
|  |  |
|  | displayWelcome(); |
|  |  |
|  | LCD\_CMD(0x01); //clear display screen |
|  | disable(); |
|  |  |
|  | } // end main |
|  |  |
|  | void displayWelcome() |
|  | { |
|  | LCD\_CMD(0x38); // 8 bit mode 2 lines |
|  | LCD\_CMD(0x0C); // display ON, cursor OFF |
|  |  |
|  | LCD\_CMD(0x80); // cursor to beginning of first row |
|  | display(msg); |
|  | LCD\_CMD(0xC0); // cursor to beginning of second row |
|  | display(msg2); |
|  | delay1(500); |
|  | } |
|  |  |
|  | void disable() |
|  | { |
|  | while(1) { |
|  | LCD\_CMD(0x80); |
|  | display(voting); |
|  | LCD\_CMD(0xC0); |
|  | display(xxx); |
|  | LCD\_CMD(0xCE); |
|  | display(DV); |
|  | GREEN = 0; |
|  | RED = 1; |
|  | if(CTRLBUTTON == 1) |
|  | enable(); |
|  | } |
|  | } |
|  |  |
|  | void enable() |
|  | { |
|  | BUZZER =1; delay1(500); |
|  | BUZZER =0; |
|  | while(1) |
|  | { |
|  | RED = 0; |
|  | GREEN = 1; |
|  | LCD\_CMD(0xCE); |
|  | display(EV); |
|  | if(VIEWSTATUS == 1) |
|  | { |
|  | viewstatus(); |
|  | } |
|  |  |
|  | if(RESULT == 1) |
|  | { |
|  | if(can1>can2 && can1>can3) |
|  | { |
|  | can1=can2=can3 =0; |
|  | LCD\_CMD(0x01); |
|  | LCD\_CMD(0x80); |
|  | display(winner1); |
|  | LCD\_CMD(0xC0); |
|  | display(congrats); |
|  | delay1(1000); LCD\_CMD(0x01); |
|  | disable(); |
|  | } |
|  | if(can2>can1 && can2>can3) |
|  | { |
|  | can1=can2=can3 =0; |
|  | LCD\_CMD(0x01); |
|  | LCD\_CMD(0x80); |
|  | display(winner2); |
|  | LCD\_CMD(0xC0); |
|  | display(congrats); |
|  | delay1(1000); LCD\_CMD(0x01); |
|  | disable(); |
|  | } |
|  | if(can3>can1 && can3>can2) |
|  | { |
|  | can1=can2=can3 =0; |
|  | LCD\_CMD(0x01); |
|  | LCD\_CMD(0x80); |
|  | display(winner3); |
|  | LCD\_CMD(0xC0); |
|  | display(congrats); |
|  | delay1(1000); LCD\_CMD(0x01); |
|  | disable(); |
|  | } |
|  | else |
|  | { |
|  | LCD\_CMD(0x01); |
|  | LCD\_CMD(0x80); |
|  | display(wrong); |
|  | LCD\_CMD(0xC0); |
|  | display(wrong2); |
|  | delay1(1000); LCD\_CMD(0x01); |
|  | disable(); |
|  | } |
|  | } |
|  |  |
|  | if(CAN1 == 1) |
|  | { |
|  | BUZZER =1; delay1(500); |
|  | BUZZER =0; |
|  | ++can1; |
|  | LCD\_CMD(0x01); |
|  | LCD\_CMD(0x80); |
|  | display(cast); |
|  | LCD\_CMD(0xC0); |
|  | display(cast1); |
|  | delay1(500); LCD\_CMD(0x01); |
|  | disable(); |
|  | } |
|  | if(CAN2 == 1) |
|  | { |
|  | BUZZER =1; delay1(500); |
|  | BUZZER =0; |
|  | ++can2; |
|  | LCD\_CMD(0x01); |
|  | LCD\_CMD(0x80); |
|  | display(cast); |
|  | LCD\_CMD(0xC0); |
|  | display(cast2); |
|  | delay1(500); LCD\_CMD(0x01); |
|  | disable(); |
|  | } |
|  | if(CAN3 == 1) |
|  | { |
|  | BUZZER =1; delay1(500); |
|  | BUZZER =0; |
|  | ++can3; |
|  | LCD\_CMD(0x01); |
|  | LCD\_CMD(0x80); |
|  | display(cast); |
|  | LCD\_CMD(0xC0); |
|  | display(cast3); |
|  | delay1(500); LCD\_CMD(0x01); |
|  | disable(); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void viewstatus() |
|  | { |
|  | itoa(can1, votes1); |
|  | LCD\_CMD(0xC0); |
|  | display(votes1); |
|  | itoa(can2, votes2); |
|  | LCD\_CMD(0xC5); |
|  | display(votes2); |
|  | itoa(can3, votes3); |
|  | LCD\_CMD(0xCA); |
|  | display(votes3); |
|  |  |
|  | delay1(500); |
|  | disable(); |
|  | } |
|  |  |
|  | //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LCD FUNCTIONS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | void display(unsigned char \*StringOfChars) |
|  | { |
|  | while (\*StringOfChars > 0) |
|  | { |
|  | Sendchar(\*StringOfChars++); |
|  | } |
|  | } |
|  |  |
|  | void Sendchar(unsigned char character) |
|  | { |
|  | LCD\_DATA(character); |
|  | } |
|  |  |
|  | void delay1(unsigned int i ) |
|  | { unsigned int d1; |
|  | unsigned char ss; |
|  | for( d1=0;d1<i;d1++) |
|  | for(ss=0;ss<255;ss++); |
|  | } |
|  |  |
|  | void LCD\_CMD(unsigned char i) |
|  | { lcd=i; |
|  | RS=0; |
|  |  |
|  | RW=0; |
|  | E=1; |
|  | delay1(1); |
|  | E=0; |
|  |  |
|  | } |
|  | void LCD\_DATA(unsigned char i ) |
|  | { |
|  | RS=1; |
|  | lcd=i; |
|  | RW=0; |
|  | E=1; |
|  | delay1(1); |
|  | E=0; |
|  | } |