NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY

Course : Microprocessor and Microcotroller Lab.

A Mini Project Report on

ELECTRONIC VOTING MACHINE USING 8051 MICROCONTROLLER

Submitted By

NAME	USN
Karthik.G.S	4NM20EC049
Kiran Poojary	4NM20EC050
Kirthik.K	4NM20EC051
Kushal.S	4NM20EC053

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

N.M.A.M. INSTITUTE OF TECHNOLOGY, NITTE -574110

2021 – 2022

ABSTARCT

In our world of technology the old methods of voting are being changed to match the trends in the other fields of life. The voting is migrating from the conventional ballot papers and boxes to the electronic methods.

This project presents a way to develop an electronic voting machine which displays the count of votes on a 16x2 LCD interface. A user can cast his/her vote through set of four switches (one for each candidate). After every cast of vote, the respective LED glows and buzzer buzzes. The voting is started by pressing the start switch after which the user is prompted to vote. The count of votes is stored in four different variables. As soon as the user votes for a candidate by pressing one of the switches, the value of the corresponding variable is increased by one. After this a Thank you message is displayed on LCD to acknowledge the registration of user's vote. The message stays on the screen until either the start button to cast another vote or Stop switch is pressed to get the poll results. When the stop button is pressed, the names of the candidates are displayed along with their vote counts. After some delay, the result is displayed which could be either declaration of the winner candidate or the candidates with a clash.

II. OBJECTIVES OF THE PROJECT

1. Interfacing 16x2 LCD display with 8051 microcontroller.

2. Interfacing set of six tactile switches with 8051 microcontroller.

3. Interfacing LED with 8051 microcontroller.

4. Interfacing buzzer with 8051 microcontroller.

III.BLOCK DIAGRAM

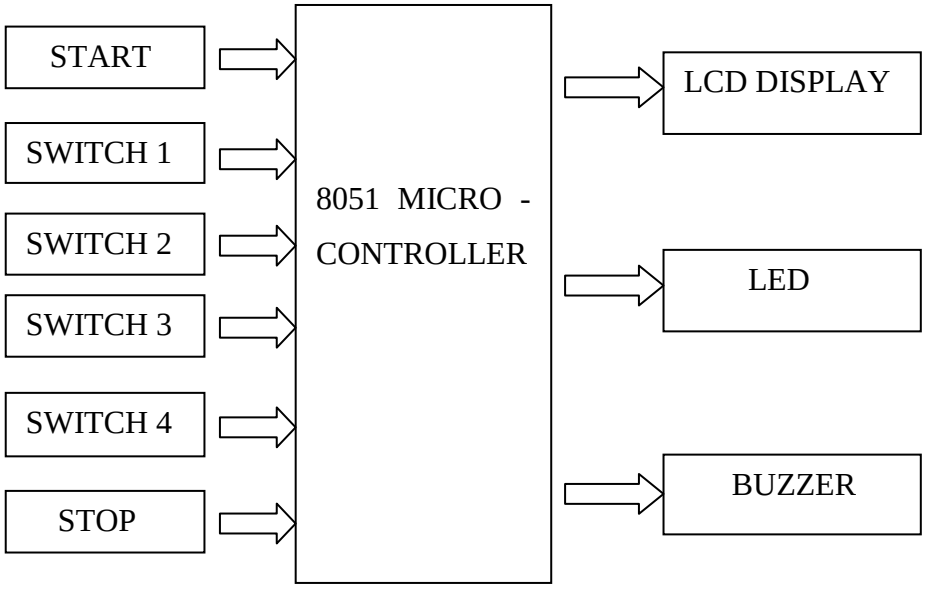

Fig 3.1 Block diagram

The LCD display is mainly used to display the result and number of votes obtained by each candidate (only when stop button is pressed), the switches are used to cast the votes and the buzzer is interfaced with microcontroller which buzzes when an vote is casted. Four LED's are also interfaced with microcontroller which are associated with four candidates, and they would glow when voter casts his/her vote to one of the candidate and the buzzer buzzes to indicate the end of the voting process

IV. Method

a. Interfacing 16x2 LCD display with 8051 microcontroller.

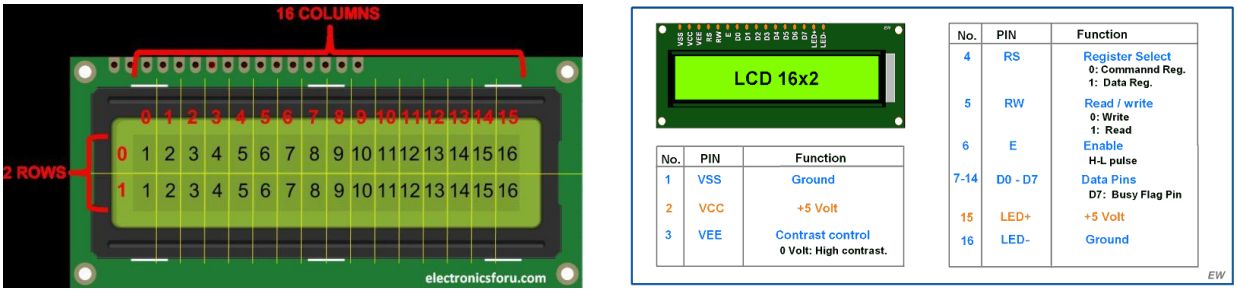

Fig 4.1 LCD display

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols.

These 16 x 2 LCD display modules are constant of 16 Columns and 2 Rows. The 1st row of this module has a total of 16 columns 0 to 15 and the position of the first row is 0. Also, the 2nd row has a total of 16 columns 0 to 15 and the position of the second row is position is 1. So the total numbers of the column are 16 x 2 = 32. Its means 16 x 2 LCD module can display 32 characters at the same time. The LCD display has got 14 pins of which pin1 and pin2 are ground and VCC, pin 3 which is used to adjust the contrast of the LCD, pin 4 is register select, pin 5 is to read and write, pin 6 is enable, pins 7-14 are data pins respectively. The data pins 7-14 are connected to port P2.0- port P2.7 of the microcontroller. The 4^{th}, 5^{th} and 6^{th} pins are connected to port P3.5, P3.6 and P3.7 of microcontroller respectively. LCD is used to display the name of the candidate and the result when the stop switch is pressed.

b. Interfacing set of six tactile switches with 8051 microcontroller.

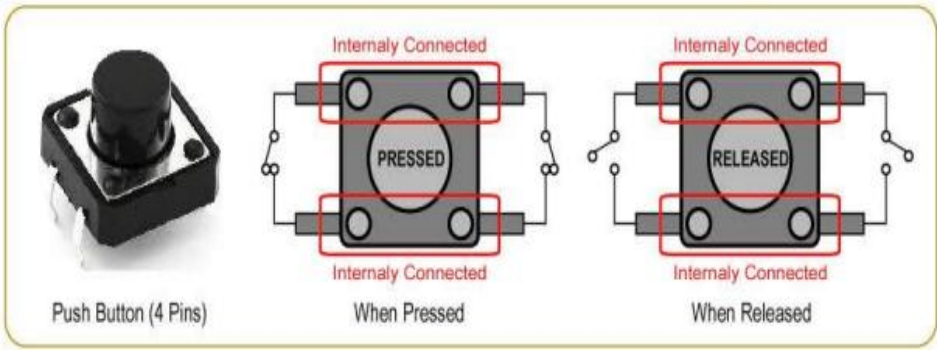

Fig 4.2 Tactical switch

A “tactile switch” is one type of widely used switch that completes an electrical circuit typically when pressure is applied to the device by the user, which then gives the user a perceptible “click” or haptic bump in response, indicating current flow. Current flow is turned off when the switch is released. To give a shorter definition, a tactile switch is a momentary action device whose operation is perceptible by touch. This tangible feedback gives some assurance to the user that the switch has operated and a signal has flowed. Tactile switch models are also available that turn off current when the switch is pushed and turn it back on when released.

Tactile switches typically contain 4 pins. These pins are internally connected into 2 sets. The purpose of using 4 pins is to provide stability when the device is mounted on a circuit board.

Tactile switches are also available with just 2 pins.

Here switch is the peripheral device interfaced with microcontroller as the input device. The voter would cast his vote to a particular candidate by pressing one of the switch. These switches are interfaced to the port P1.1, P1.2, P1.3, P1.4 of the controller. There are two more switches (start and stop) both being active low is connected to the P1.0 and P1.5 of the microcontroller used to start and terminate the process.

c. Interfacing LED with 8051 microcontroller.

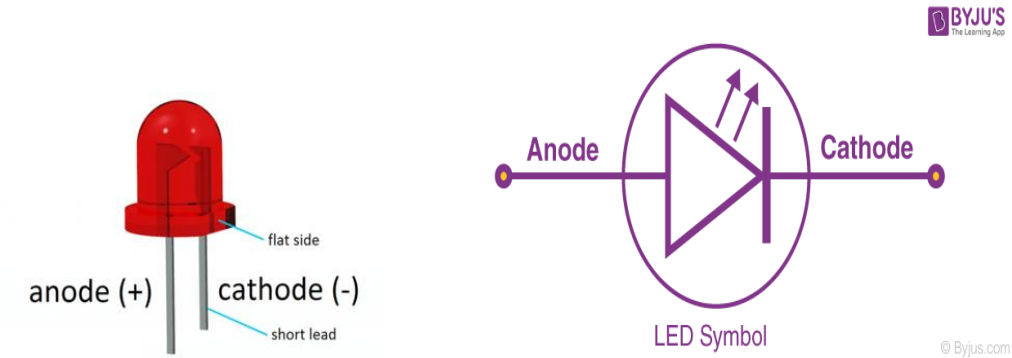

Fig 4.3 Light Emmiting Diode

A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it. When current passes through an LED, the electrons recombine with holes emitting light in the process. LEDs allow the current to flow in the forward direction and blocks the current in the reverse direction.

Light-emitting diodes are heavily doped p-n junctions. Based on the semiconductor material used and the amount of doping, an LED will emit coloured light at a particular spectral wavelength when forward biased. As shown in the figure, an LED is encapsulated with a transparent cover so that emitted light can come out.

There are four LED's interaced with port pins P0.0, P0.1, P0.2, P0.3 of the microcontroller. These four LED's are assigned to four candidates. When the voter casts his vote, the associated LED would glow.

d. Interfacing buzzer with 8051 microcontroller.

Fig 4.4 Buzzer

The pin configuration of the buzzer is shown in Fig 4.4. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-'symbol or short terminal and it is connected to the GND terminal.

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren

The buzzer is interfaced to the port pin P0.7 of the microcontroller. The buzzer has got two terminals namely positive and negative which are connected to VCC and P0.7 respectively. The buzzer is interfaced so that voter can make sure that he/she has casted thier votes. Whenever voter votes to a particular candidate through tactile switches, the buzzer would buzz for some time indicating that the vote has been casted.

V. Result

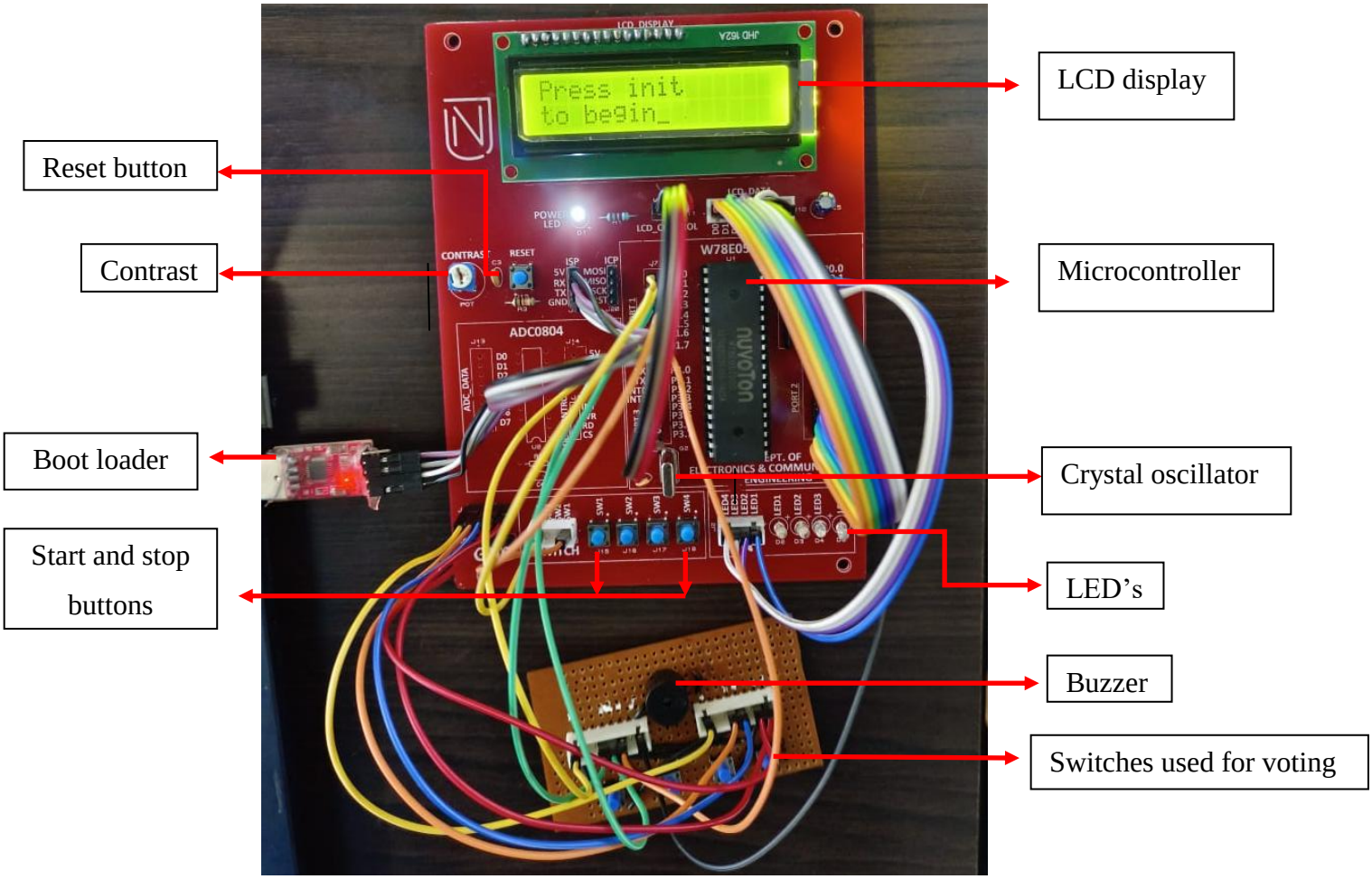

Fig 5.1 Project setup

Discussion:

1. When the power is switched on, the following message is displayed on the LCD:

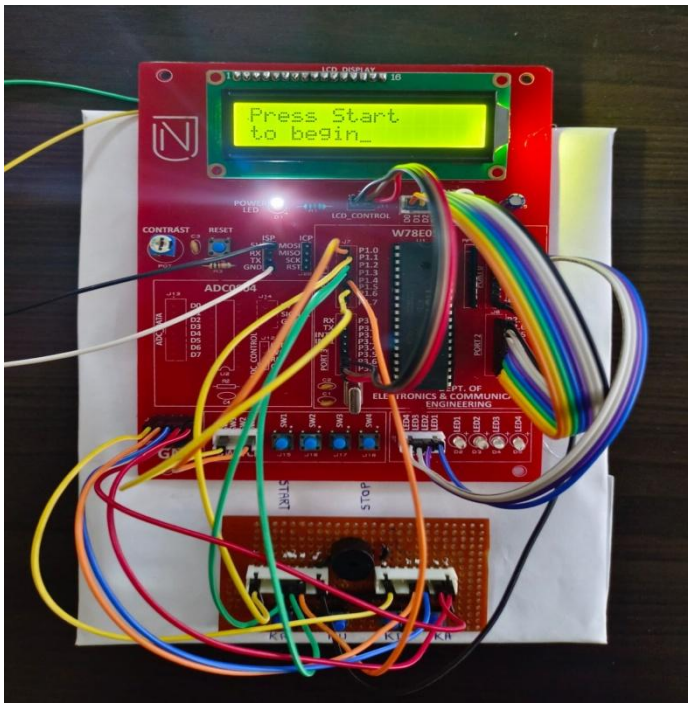

Fig 5.2 Setup when power Is turned on

2. After the start button is pressed, the following message is displayed on the LCD

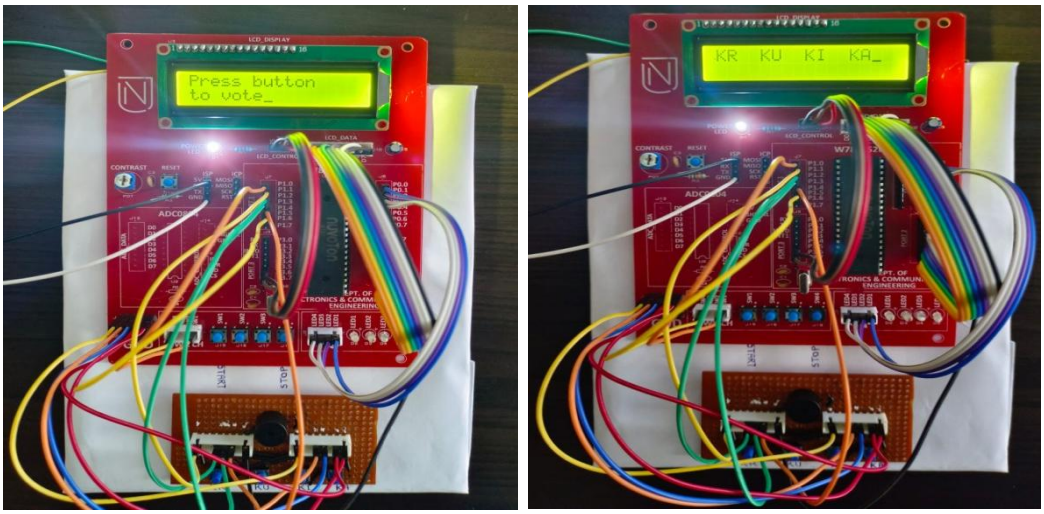

Fig 5.3 Setup after start button is pressed.

3. After the voter votes for the particular candidate, the following message is displayed on the LCD:

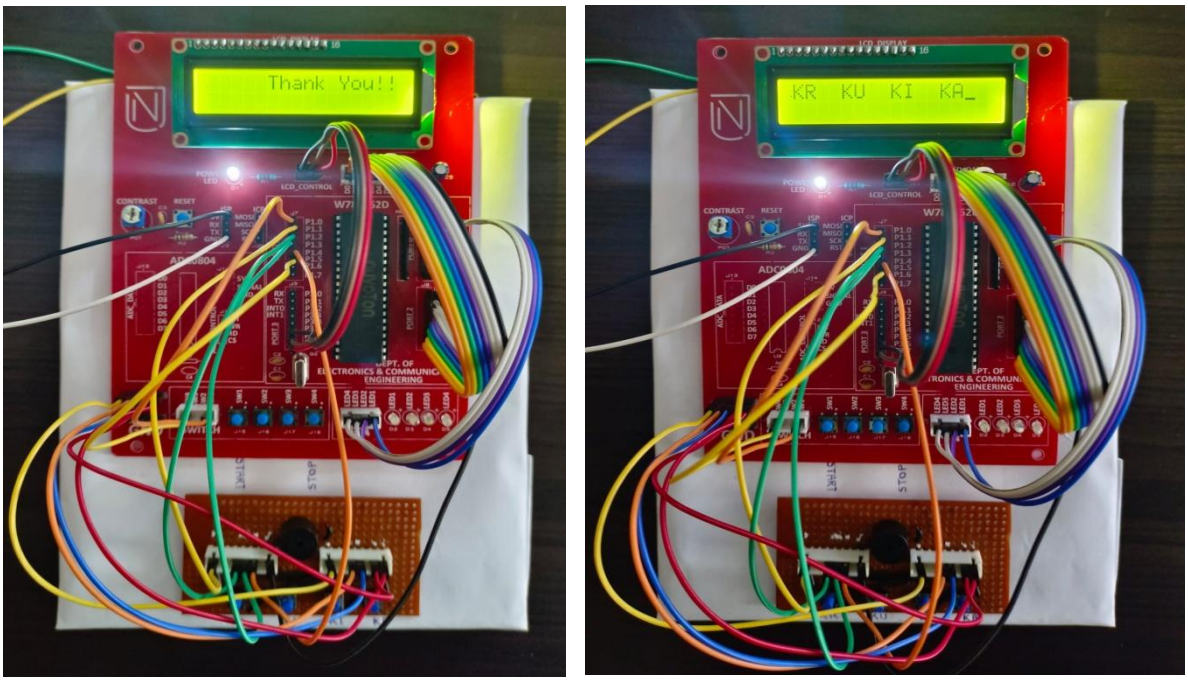

Fig 5.3 Setup after voter votes.

4. After the stop button is pressed , the number of votes obtained by each candidate is displayed on the lcd and the winner or clash between the candidates is displayed on the display.

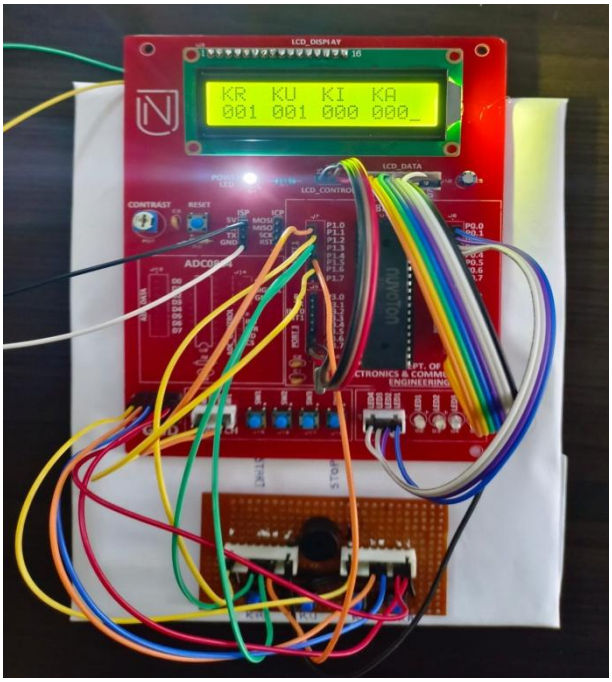

Fig 5.4 Setup after stop button is pressed

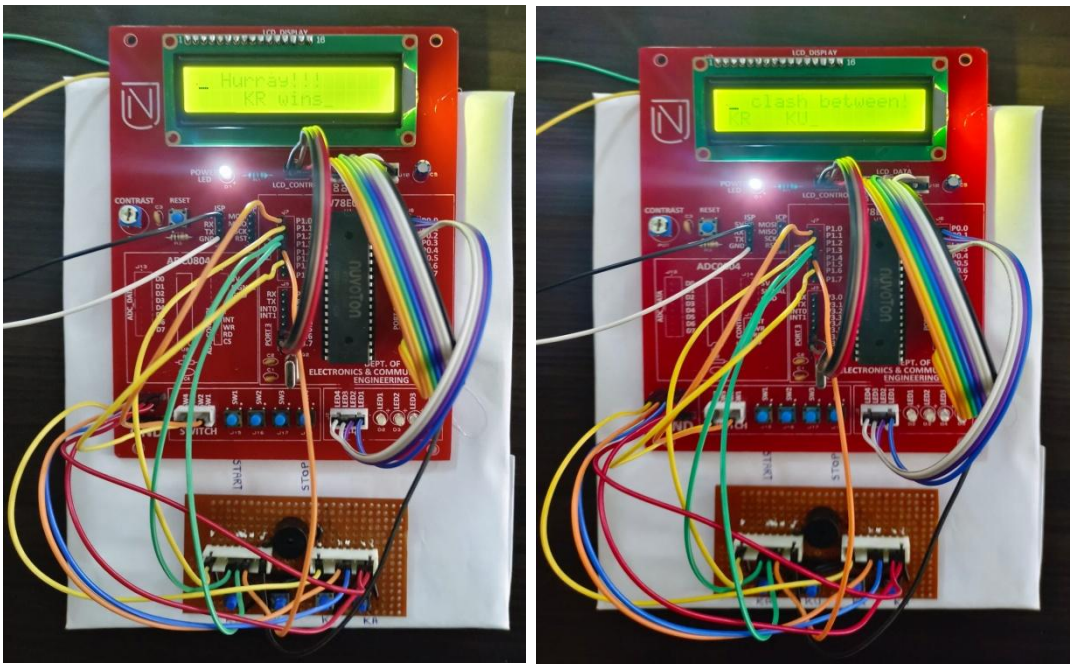

Fig 5.5 Announcement of result.

Code snippet:

```
#include<reg51.h>

#define msec 50

#define lcd_data_str_pin P2

sbit rs = P3^5;  //Register select (RS) pin

sbit rw = P3^6;  //Read write(RW) pin

sbit en = P3^7;  //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 led_1=P0^0;

sbit candidate_2=P1^2;  //Candidate2

sbit led_2=P0^1;

sbit candidate_3=P1^3;  //Candidate3

sbit led_3=P0^2;

sbit candidate_4=P1^4;  //Candidate4

sbit led_4=P0^3;

sbit buzzer=P0^7;

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;


led_1=1;led_2=0;led_3=0;led_4=0;

                        buzzer=0;

                        delay(250);


led_1=0;led_2=0;led_3=0;led_4=0;

                        buzzer=1;

  }

}


if (candidate_2==1)

{

 while (candidate_2 == 1);

  {

   vote_2 = vote_2 + 1;


led_1=0;led_2=1;led_3=0;led_4=0;

                         buzzer=0;

                        delay(250);


led_1=0;led_2=0;led_3=0;led_4=0;

                         buzzer=1;

  }

}


if (candidate_3==1)


{

 while (candidate_3 == 1);

  {

   vote_3 = vote_3 + 1;


            led_1=0;led_2=0;led_3=1;led
_4=0;

                        buzzer=0;


            delay(250);


led_1=0;led_2=0;led_3=0;led_4=0;

                        buzzer=1;

  }

}


if (candidate_4==1)


{

 while (candidate_4 == 1);

  {

   vote_4 = vote_4 + 1;


            led_1=0;led_2=0;led_3=0;led
_4=1;

                        buzzer=0;


            delay(250);


led_1=0;led_2=0;led_3=0;led_4=0;

                        buzzer=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("KR");
delay(msec);
lcd_cmd(0x84);
delay(msec);
lcd_data_str("KU");
delay(msec);
lcd_cmd(0x88);
delay(msec);
lcd_data_str("KI");
delay(msec);
lcd_cmd(0x8C);
delay(msec);
lcd_data_str("KA");
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("KR");
delay(msec);
lcd_cmd(0x84);
delay(msec);
lcd_data_str("KU");
delay(msec);
lcd_cmd(0x88);
delay(msec);
lcd_data_str("KI");
delay(msec);
lcd_cmd(0x8C);
delay(msec);
lcd_data_str("KA");
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(500);
```

```
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("KR");

 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("KU");

 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("KI");

 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("KA");

 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_1 == max)
        {
                lcd_cmd(0xC0);
 lcd_data_str("KR");
 delay(200);
        }
 if(vote_2 == max)
 {
  lcd_cmd(0xC5);
  lcd_data_str("KU");
  delay(200);
 }
 if(vote_3 == max)
 {
  lcd_cmd(0xC9);
  lcd_data_str("KI");
  delay(200);

}
 }
 if(vote_4 == max)
 {
  lcd_cmd(0xCD);
  lcd_data_str("KA");
  delay(200);
 }
}
}

void main()
{
led_1=led_2=led_3=led_4=0;
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("start");
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();
}
```