// LCD based voting machine using LCD

#include<reg51.h>

#define msec 50

#define lcd\_data\_str\_pin P0

sbit rs = P2^7; //Register select (RS) pin

sbit rw = P2^6; //Read write(RW) pin

sbit en = P2^5; //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 candidate\_2=P1^2; //Candidate2

sbit candidate\_3=P1^3; //Candidate3

sbit candidate\_4=P1^4; //Candidate4

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;

}

}

if (candidate\_2==1)

{

while (candidate\_2 == 1);

{

vote\_2 = vote\_2 + 1;

}

}

if (candidate\_3==1)

{

while (candidate\_3 == 1);

{

vote\_3 = vote\_3 + 1;

}

}

if (candidate\_4==1)

{

while (candidate\_4 == 1);

{

vote\_4 = vote\_4 + 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("P1");

delay(msec);

lcd\_cmd(0x84);

delay(msec);

lcd\_data\_str("P2");

delay(msec);

lcd\_cmd(0x88);

delay(msec);

lcd\_data\_str("P3");

delay(msec);

lcd\_cmd(0x8C);

delay(msec);

lcd\_data\_str("P4");

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("P1");

delay(msec);

lcd\_cmd(0x84);

delay(msec);

lcd\_data\_str("P2");

delay(msec);

lcd\_cmd(0x88);

delay(msec);

lcd\_data\_str("P3");

delay(msec);

lcd\_cmd(0x8C);

delay(msec);

lcd\_data\_str("P4");

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(300);

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("P1");

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("P2");

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("P3");

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("P4");

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\_2 == max)

{

lcd\_cmd(0xC5);

lcd\_data\_str("P2");

delay(50);

}

if(vote\_3 == max)

{

lcd\_cmd(0xC9);

lcd\_data\_str("P3");

delay(50);

}

if(vote\_4 == max)

{

lcd\_cmd(0xCD);

lcd\_data\_str("P4");

delay(50);

}

}

}

void main()

{

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("init");

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();

}

}