

EXPERIMENT NO. 02

ELECTRONIC VOTING MACHINE USING 8051.

DOP:

DOS:

Project Members:

- | | |
|--------------------|----|
| 1) Prasad Pawaskar | 58 |
| 2) Vishal Thakur | 72 |

AIM: To implement Electronic voting machine using 8051 microcontroller.(AT89C51)

Tools: Keil uVision 4, ISIS Proteus 7

Theory: Electronic voting machine has now days become an effective tool for voting. It ensures flawless voting and thus has become more widespread. It ensures people about their vote being secured. It avoids any kind of malpractice and invalid votes. Also such kind of system becomes more economical as consequent expenditure incurred on manpower is saved. It is also convenient on the part of voter, as he has to just press one key whichever belongs to his candidates.

voting machines are the total combination of mechanical, electromechanical, or electronic equipment (including software, firmware, and documentation required to program control, and support equipment), that is used to define ballots; to cast and count votes; to report or display election results; and to maintain and produce any audit trail information. The first voting machines were mechanical but it is increasingly more common to use electronic voting machines.

A voting system includes the practices and associated documentation used to identify system components and versions of such components; to test the system during its development and maintenance; to maintain records of system errors or defects; to determine specific changes made after initial certification; and to make available any materials to the voter (such as notices, instructions, forms, or paper ballots).

Traditionally, a voting machine has been defined by the mechanism the system uses to cast votes and further categorized by the location where the system tabulates the votes.

Voting machines have different levels of usability, security, efficiency and accuracy. Certain systems may be more or less accessible to all voters, or not accessible to those voters with certain types of disabilities. They can also have an effect on the public's ability to oversee elections.

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density non-volatile memory technology and is compatible with the industry-standard MCS-51

instruction set and pin-out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

- Algorithm m:
- 1) Initialize ports & LCD display.
 - 2) Then, initialize delay.
 - 3) Test whether the which key is pressed and find the count of votes.
 - 4) Function to display results.
 - 5) Also, display the error message if unexpected result is obtained.

Code: **// Code of Electronic Voting Machine**

```
#include<reg51.h>
#define msec 50
#define lcd_data_str_pin P2
sbit rs = P3^0; //Register select (RS) pin
sbit rw = P3^1; //Read write(RW) pin
sbit en = P3^6; //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);

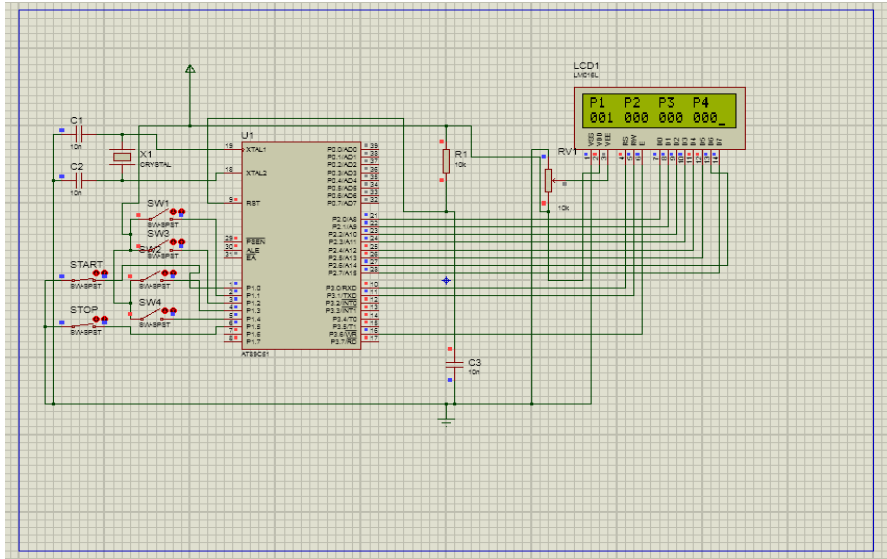
```

```
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();
}
}
```

Output :



Conclusion: Thus, an Electronic Voting Machine using AT89C51 was simulated using proteus with the help of keil. The results of voting were virtually verified using LCD in proteus.