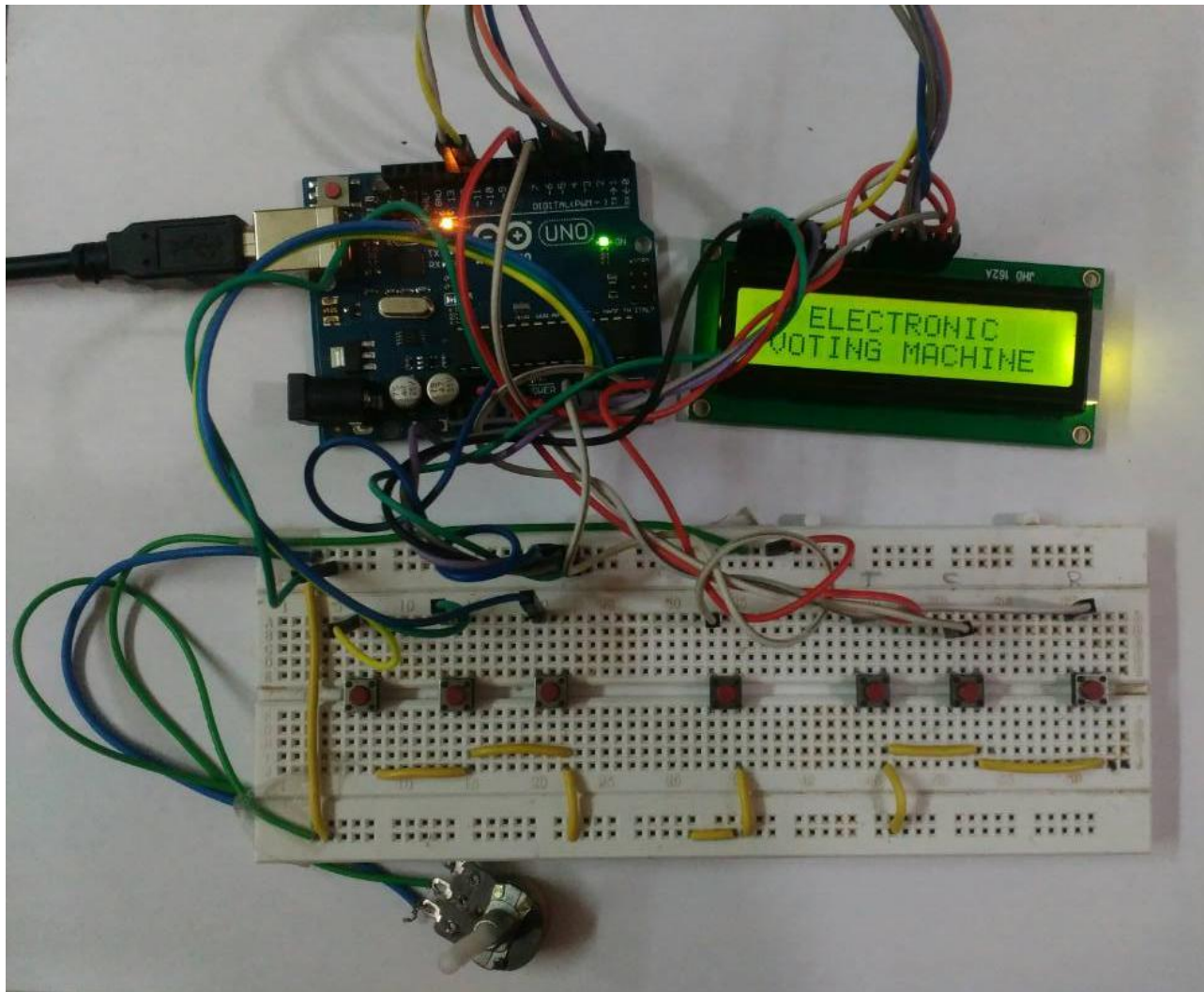


ELECTRONIC VOTING MACHINE using ARDUINO UNO



Team Members:

Bhavani B M (1PE15CS038)

G L Priya (1PE15CS054)

Akshita M (1PE15CS016)

AIM:

To design and implement Electronic Voting Machine using Arduino UNO which stores, calculates and displays the result of the votes casted by the people.

INTRODUCTION:

Our project aims at the implementation of an Electronic Voting Machine (EVM) using Arduino Uno that has several features over present voting machine.

Key features of our project are:

- Every step of polling operation is under control of the supervisor.
- Poll results can be obtained instantly.
- Voter will be aware of the candidate he is voting for

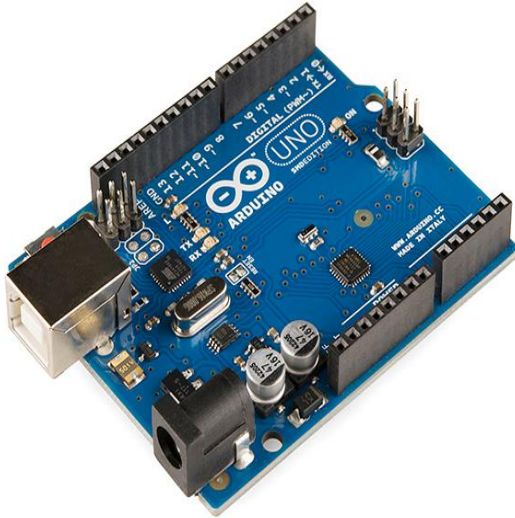
HARDWARE Specifications:

1. Arduino UNO: 1
2. 16x2 LCD: 1
3. Push Buttons: As required
4. Potentiometer 100k ohm: 1
5. Bread Board: 1
6. Power supply
7. Connecting wires: As required

SOFTWARE Specification:

ARDUINO IDE(version used: 1.8.2)

ARDUINO UNO

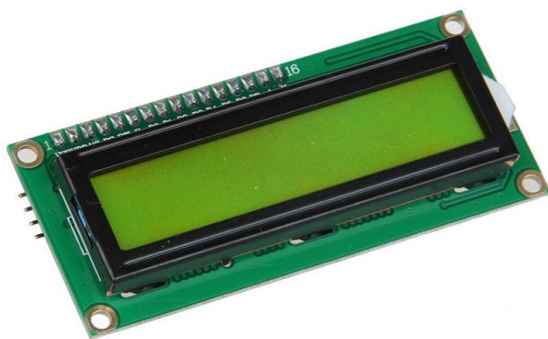


Arduino Uno is a microcontroller board based on the ATmega328P .

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

LCD (Liquid Crystal Display)



LCD screen is an electronic display module

LCD 16x2 means it can display 16 characters per line and there are such 2 lines in this LCD

In this 16x2 LCD we have 16 pins which should be connected to Arduino

The 16 pins are:

1: Vss	4: RS	7-14: [Data Bits(0-7)]
2: Vdd	5: R/W	15: A(LED+)
3: Constrast	6: E	16: K(LED-)

Abstract

Electronic Voting Machine is a simple electronic device used to record votes in place of ballot papers and ballot box which are used in earlier voting system. This is a long time consuming process and much prone to errors. In earlier elections be it state election or central election a voter used to cast his/her name and then folding the ballot paper as per prescribed method before putting it in the ballot box. This situation will continued till election scene was completely changed by electronic voting machine. No more ballot paper, ballot papers, stamping etc. All this condensed into simple box called ballot unit of electronic voting machine.

DESCRIPTION

Working :

The electronic voting machine has two parts:

- *Control Unit* : It stores all the votes in the memory and is generally battery operated. Moreover it also contains important buttons like result, close, print, clear.
- *Ballot Unit* : It contains buttons of the contesting parties. A connecting cable is used for connecting ballot unit with the control unit.

Three buttons are provided for three contesting parties (PARTY-A , PARTY-B, PARTY-C) . Voters can use these buttons to vote for their party. So these three buttons are accessible to the voters. Voters are not given the access of the rest of the four buttons.

Total button can be pressed any time to get the count of the total number of votes that are polled so far.

Close button if pressed closes the voting process and if this button is pressed then PARTY-A, B and C buttons will become ineffective and pressing of these three buttons will not have any effect on the system.

Once Close button is pressed, Result button can be pressed to check the results of the poll. Once the Result button is pressed the machine will tell you the exact number of votes that are polled for each of the three contesting parties and it will also tell the result of the poll that is which party has won the election.

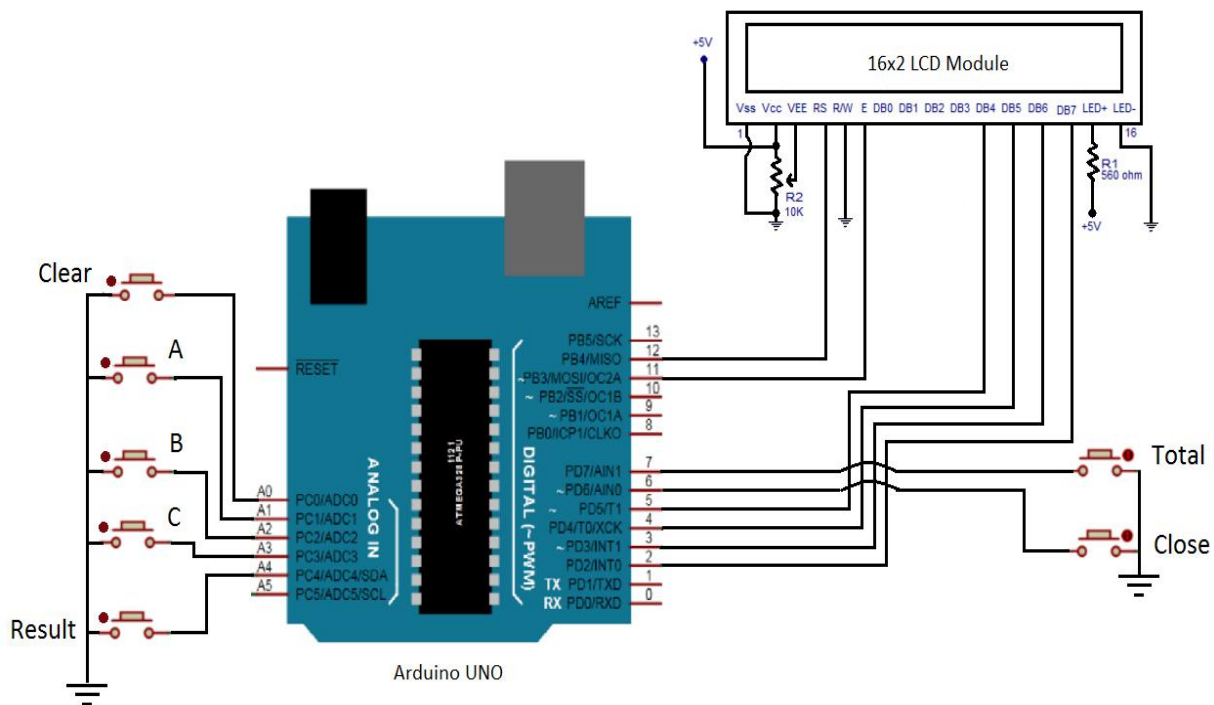
The machine is programmed to also indicate the vote difference between the winning party and the first runner up party. In case a tie happens between any two parties, or between all three contesting parties then the machine will show the tie result on the LCD screen.

Clear button is also provided and if this button is pressed then entire memory of the machine gets empty and all the stored vote count becomes zero.

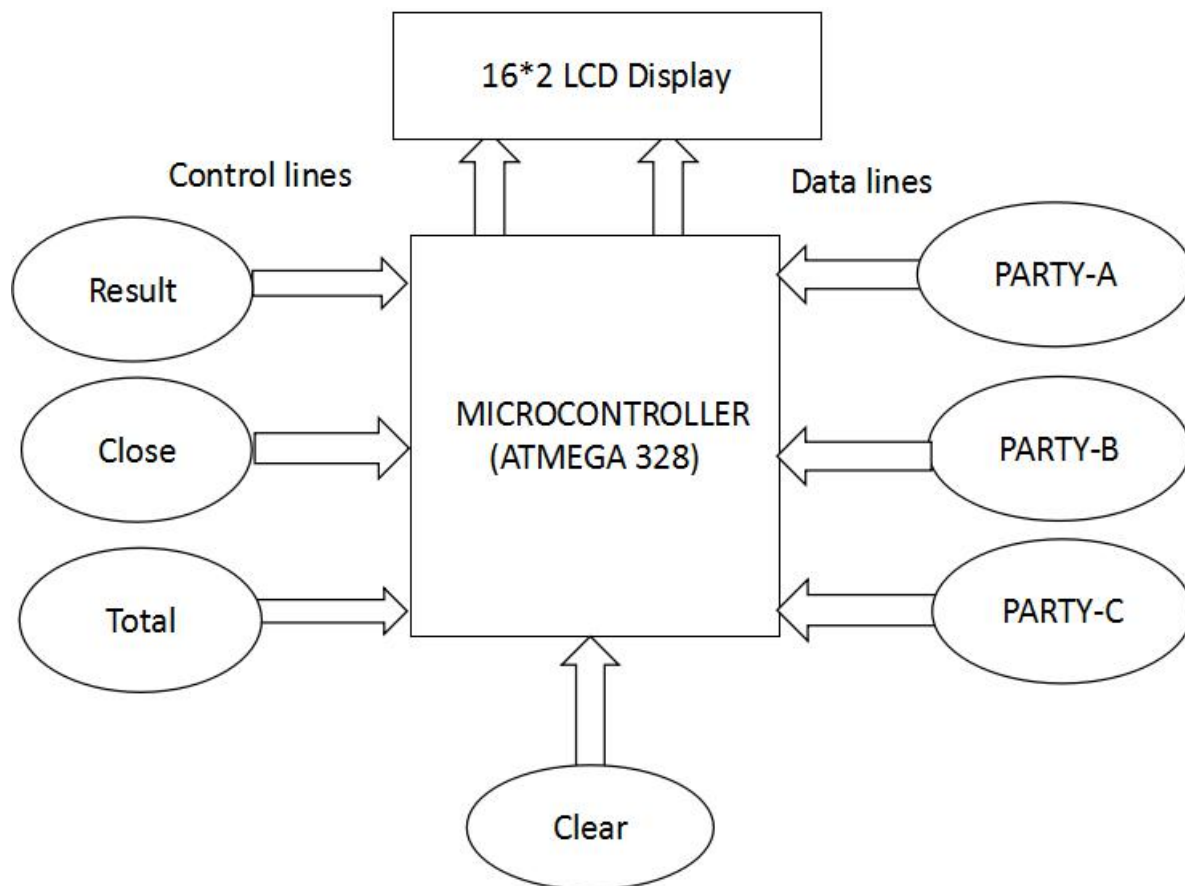
So clear and total button can be pressed any time during the voting process and if these buttons are pressed corresponding action that is programmed into the system will happen. Close button must be pressed before result button is pressed to get the result.

The vote count is saved in the on chip EEPROM of the Arduino and so if power cut occurs then also the votes that are entered into the system will remain safe and again when the system is powered up the voting will begin from the previous state.

CIRCUIT Diagram



Block Diagram



CODE:

*/*This code accepts the votes obtained from the EVM, computes the total, displays the winning party along with the number of votes by which it has won, on the Liquid Crystal Display*/*

*/*Using EEPROM, the polling data is saved even if the power supply is interrupted and also clears the memory when the clear button is pressed*/*

```
#include<LiquidCrystal.h> // lcd library
```

```
#include <EEPROM.h>      //EEPROM library
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // LCD control and data lines
```

```
// button pin definitions
```

```
int result=A4;
```

```
int closed=6;
```

```
int total=7;
```

```
int clearbutton=A0;
```

```
int a=A1;
```

```
int b=A2;
```

```
int c=A3;
```

```
// temporary variables used in program
```

```
int v1;
```

```
int v2;
```

```
int v3;
```

```
int v;
```

```
int cflag;
```

```

void setup()
{
    cflag=EEPROM.read(0);  //read the status of cflag from memory
    v1=EEPROM.read(1);    // v1 stores the vote count of party A
    v2=EEPROM.read(2);    // v2 stores the vote count of party B
    v3=EEPROM.read(3);    // v3 stores the vote count of party C

    // declaration of buttons as input
    pinMode(a,INPUT);
    pinMode(b,INPUT);
    pinMode(c,INPUT);
    pinMode(total,INPUT);
    pinMode(closed,INPUT);
    pinMode(result,INPUT);
    pinMode(clearbutton,INPUT);

    //default status of buttons when not pressed is high
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(total,HIGH);
    digitalWrite(closed,HIGH);
    digitalWrite(result,HIGH);
    digitalWrite(clearbutton,HIGH);
    lcd.begin(16,2);

```



```
if(cflag==0)
{
    lcd.print("  ELECTRONIC");
    lcd.setCursor(0,1);
    lcd.print(" VOTING MACHINE");
    delay(5000);
    lcd.clear();
    lcd.print("  Press button");
    lcd.setCursor(0,1);
    lcd.print("  to vote...");
    delay(1000);
}
```

```
if(cflag==1)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("  Voting Closed");
    lcd.setCursor(0,1);
    v=v1+v2+v3;
    lcd.print("Total Votes:");
    lcd.print(v);
}
}
```

```
void rpt()
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Press button");
    lcd.setCursor(0,1);
    lcd.print(" to vote...");
}
```

// function to calculate vote difference

```
void votedifference()
{
    if(v1>v2)
    {
        if(v2>v3)
        {
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("A wins by");
            lcd.setCursor(0,1);
            lcd.print(v1-v2);
            lcd.print(" votes");
        }
        else
        {
```

```
if(v1>v3)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("A wins by");
    lcd.setCursor(0,1);
    lcd.print(v1-v3);
    lcd.print(" votes");
}
```

```
if(v1<v3)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("C wins by");
    lcd.setCursor(0,1);
    lcd.print(v3-v1);
    lcd.print(" votes");
}
}
```

```
else
{
    if(v1>v3)
    {
```

```
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("B wins by");  
lcd.setCursor(0,1);  
lcd.print(v2-v1);  
lcd.print(" votes");  
}
```

else

```
{  
  if(v2>v3)  
  {  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("B wins by");  
    lcd.setCursor(0,1);  
    lcd.print(v2-v3);  
    lcd.print(" votes");  
  }
```

```
  if(v2<v3)  
  {  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("C wins by");  
    lcd.setCursor(0,1);
```

```
        lcd.print(v3-v2);  
        lcd.print(" votes");  
    }  
}  
}
```

```
void loop()  
{  
// if party A button is pressed  
    if(digitalRead(a)==LOW && cflag==0)  
    {  
        v1=v1+1;  
        EEPROM.write(1,v1);  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("Vote received...");  
        lcd.setCursor(0,1);  
        lcd.print(" Thank You!!");  
        delay(1500);  
        rpt();  
    }
```

```
// if party B button is pressed  
    if(digitalRead(b)==LOW && cflag==0)  
    {
```

```
v2=v2+1;
EEPROM.write(2,v2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Vote received...");
lcd.setCursor(0,1);
lcd.print(" Thank You!!");
delay(1500);
rpt();
}
```

// if party C button is pressed

```
if(digitalRead(c)==LOW && cflag==0)
{
v3=v3+1;
EEPROM.write(3,v3);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Vote received...");
lcd.setCursor(0,1);
lcd.print(" Thank You!!");
delay(1500);
rpt();
}
```

// if total button is pressed but close button was not pressed earlier

```
if(digitalRead(total)==LOW && cflag==0)
{
    v=v1+v2+v3;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Total votes:");
    lcd.setCursor(0,1);
    lcd.print(v);
    delay(2000);
    rpt();
}
```

// if close button is pressed

```
if(digitalRead(closed)==LOW)
{
    cflag=1;
    EEPROM.write(0,cflag);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Voting Closed!!!");
    lcd.setCursor(0,1);
    v=v1+v2+v3;
    lcd.print(" Total Votes:");
    lcd.print(v);
```

```
while(digitalRead(result)==HIGH);  
}
```

```
// if result button is pressed after the close button was pressed
```

```
if(digitalRead(result)==LOW && cflag==1)  
{  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("A:");  
    lcd.print(v1);  
    lcd.setCursor(7,0);  
    lcd.print("B:");  
    lcd.print(v2);  
    lcd.setCursor(0,1);  
    lcd.print("C:");  
    lcd.print(v3);  
    delay(3000);
```

```
// logic for result of voting process
```

```
if(v1==v2 && v2==v3)  
{  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print(" Result Tied!!!");  
    delay(3000);  
}
```



```
if(v1==v2 && v1>v3)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Tie b/w A and B");
    delay(3000);
}
```

```
if(v2==v3 && v2>v1)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Tie b/w B and C");
    delay(3000);
}
```

```
if(v1==v3 && v1>v2)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Tie b/w A and C");
    delay(3000);
}
```

```
if(v1>v2)
{
```

```
if(v1>v3)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Party A wins!!!");
    delay(3000);
    votedifference();
}
```

```
else if(v3>v1)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Party C wins!!!");
    delay(3000);
    votedifference();
}
}
```

```
else
{
    if(v2>v3 && v1!=v2)
    {
        lcd.clear();
        lcd.setCursor(0,0);
```

```

    lcd.print(" Party B wins!!!");
    delay(3000);
    votedifference();
}

else if(v3>v2)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Party C wins!!!");
    delay(3000);
    votedifference();
}
}
}

```

// if clear button is pressed

```

if(digitalRead(clearbutton)==LOW)
{
    for (int i = 0; i < 512; i++)
    {
        EEPROM.write(i, 0);
    }
    v1=0;
    v2=0;
    v3=0;
}

```

```
cflag=0;

lcd.clear();

lcd.setCursor(0,0);
lcd.print(" Memory Cleared");

lcd.setCursor(0,1);
lcd.print(" ***** ");

delay(3000);

lcd.clear();

  lcd.setCursor(0,0);
  lcd.print(" Press button");
  lcd.setCursor(0,1);
  lcd.print(" to vote...");
}
}

/*END OF PROGRAM*/
```

Expected Outcome:

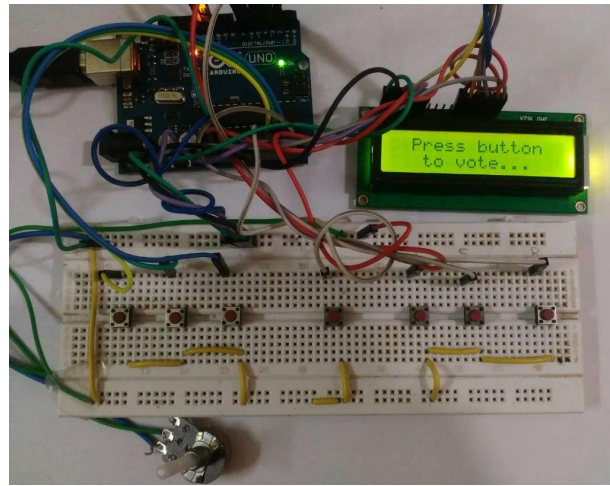
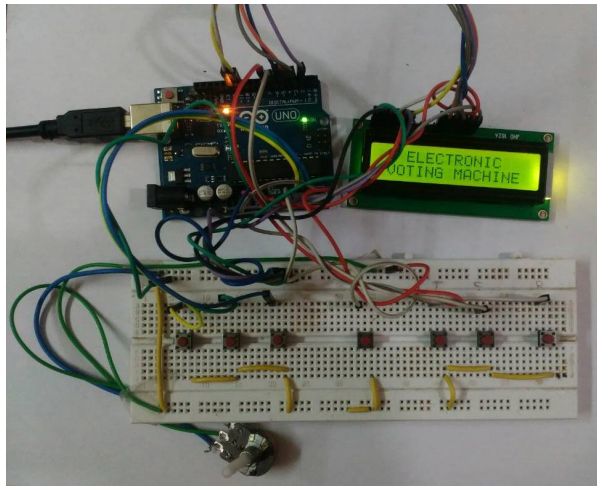
- In this project, we intend to observe that the vote is polled only to that Party that the individual wants to, and not to any other Party as a malfunction of the Prototype.
- Special attention is provided to the Control unit buttons namely, the Total, Close, Result and Clear button that are visible only to the supervisor and not to the voters and the working of each of these buttons is tested with all possible test cases to ensure absence of any data manipulation.
- The Close button should close the voting process and the EVM should not consider any further voting.
- The Total count of votes for each of the parties are to be governed in a proper manner and extreme cases such as ties between the parties are also monitored such that when the result button is pressed, the parties and their corresponding votes is displayed, followed by the winning party and the vote difference between the winning party and the runner-up party.
- The Clear button should clear the memory of the EVM such that the prototype can be used for a new Voting process.

Output:

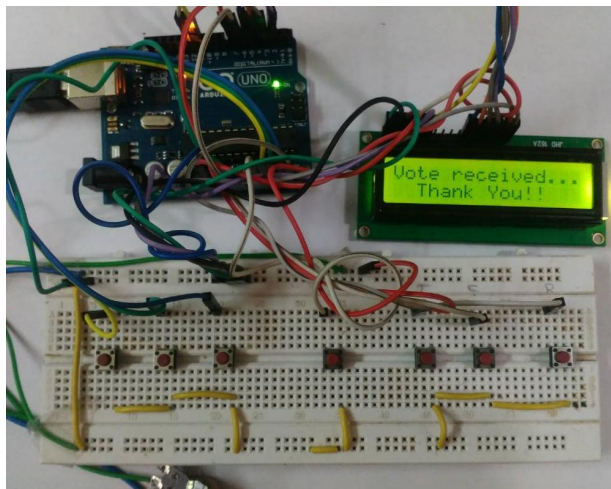
The following are some of the snapshots of the prototype when various buttons are pressed.

The first three push buttons refer to Party-A, Party-B and Party-C respectively. The fourth button refers to Clear and the fifth, sixth and seventh button refers to Total, Close and Result respectively pertaining to the voting process.

Initially:

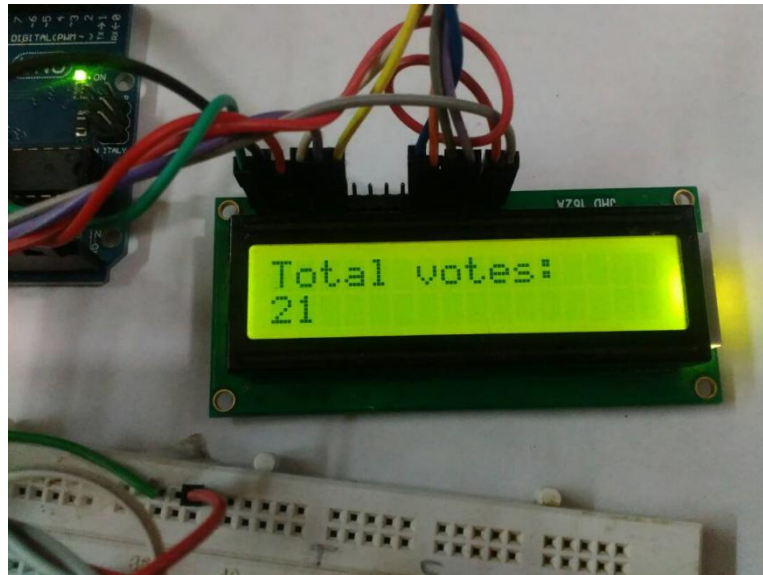


When a vote is casted:

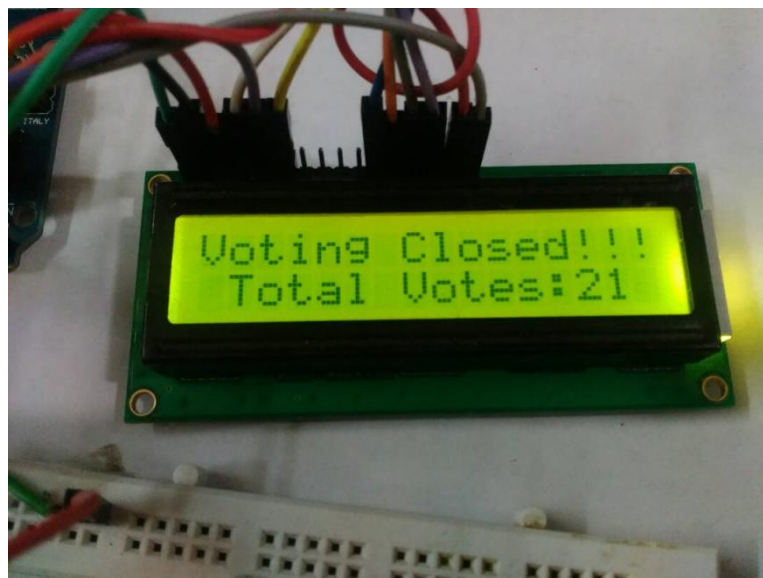


After subsequent voting:

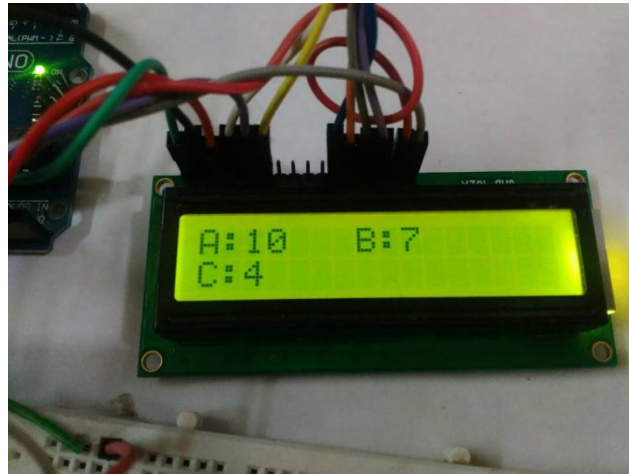
On pressing Total button:



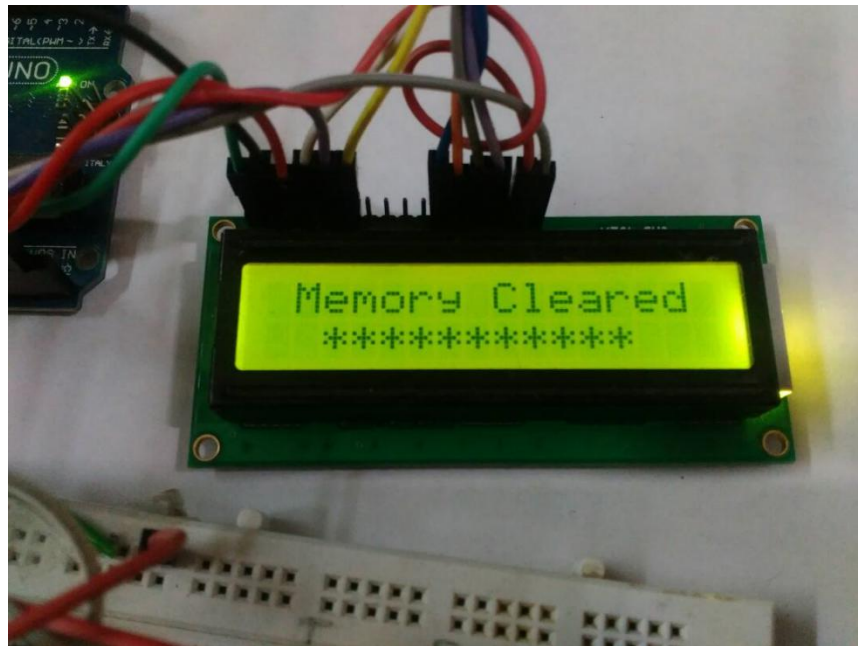
On pressing close button:



On pressing Result button:



When Clear button is pressed:



Beginning of a new Voting process:



Future Scope:

This project can be further extended in the following ways:

- Interfacing a GSM modem with the machine for sending SMS to the mobile phone for all the participating candidates/parties so that they can also get to know the results of each and every voting machine instantly
- A Timer could be included, which could automatically end the voting after specified duration of time.
- Biometric Verifications of voters, so that automatically it can be ensured that one person is voting once only.
- EVM can be made more interactive by adding Sound Effects (speech) to it.