VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ELECTRONIC VOTING MACHINE

AYUSH SINGH
19BEC0754

Electronics and Communication

Vellore Institute of Technology Vellore, India

ATHARV ARYA
19BEC0758

Electronics and Communication

Vellore Institute of Technology Vellore, India

AMAN LAKHMANI
19BEC0787

Electronics and Communication

Vellore Institute of Technology Vellore, India

***Abstract*—The Electronic Voting Machine (EVM) is a simple electronic device used to record votes in place of ballot papers and boxes which were used before in conventional voting system. Electronic voting machine has now replaced the traditional voting system due to several advantages like security, automatic counting etc.**

In this project we describe a microcontroller based Electronic Voting Machine. This system is constructed using 8051 microcontrollers (AT89C51), The circuit is constructed using proteus software and its software program is written with C language using Keil software. This is an effective tool for voting, this system accepts votes and keep counting the total votes polled and LCD is used to display vote conformation and election results, by pressing the result switch the subsequent count can be seen on LCD.

I. INTRODUCTION

Electronic Voting Machine known as EVM in modern days became a powerful tool in Indian election for voting, known for it's security and flawless ease of voting. Use of EVMs instead of traditional ballot voting, it drastically reduces the overall cost as consequent expenditure incurred on manpower is saved. It is also easy for voters as one only have to press the button which represents their interested party. Voting machines are the total combination of mechanical, electromechanical, or electronic equipment that is used to define ballots, to cast and count votes, to report or display election results. Electronic voting technology intends to speed the counting of ballots, reduce the cost of paying staff to count votes manually and can provide improved accessibility for disabled voters. Also in the long term, expenses are expected to decrease. Results can be reported and published faster. 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, accuracy and efficiency. 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 microcontroller used in this project 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 standardmcs-51 instruction set and pinout. 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.

This work presents Electronic Voting Machine (EVM) using 8051 Microcontroller. Electronic Voting Machine (EVM) is a simple electronic device used to record votes in place of ballot papers and boxes which were used before in conventional voting system.

Electronic voting machine has now replaced the traditional voting system due to several advantages like security, automatic counting etc. The simulation result is provided by using proteus, the goal is to understand how microcontroller works and its application.

II. LITERATURE SURVEY

Shaw et al.[1] The authors proposes a voting system that provides dual verification of the voters that requires their fingerprint and unique voter id The process of verification involves matching of this id and fingerprint from the database. This is a faster and more secured way of holding elections. Our system is secured, reliable and also cost-effective.

Priya et al.[2] It proposes a system that includes a multiple layers of verification to ensure the reliability of the device by using biometric fingerprint sensor.

Once the corresponding fingerprint is matched with the information provided, the voter will be allowed to proceed for choosing their preferred candidate from the panel of buttons. The final vote is then displayed onto a LCD for the satisfaction of voters. The proposed project displays transparency and also carries the feature of being autonomous during the course of operation.

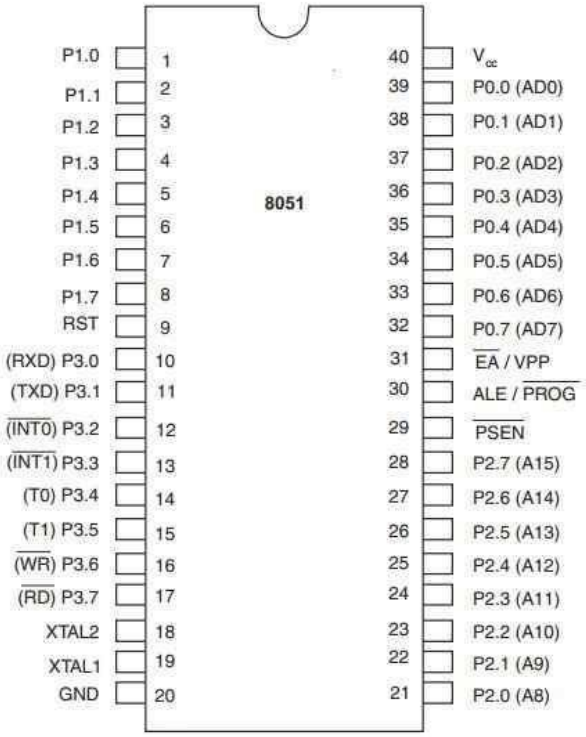
Rasaily et al.[3] This project describes a microcontroller-based Electronic Voting Machine which can be used in real time systems. The system is constructed using AT89C51 (microcontroller), Voltage Regulator 7805 and its software program is written with assembly language. It is an effective tool for voting ,it is a combination of mechanical electromechanical and electronic equipment and is used to display election results and to maintain and produce any audit trial information which can be used in real time systems.

Ali et al.[4] An electronic voting machine EVM is introduced in this paper which replaced conventional methods of voting i.e manual voting. The Proposed machine in this paper is faster, efficient, and reliable and error free as compared to manual voting system which is slower, poses full day fatigue on people and chances of error are greater.

Balzarotti et al.[5] In this paper, author describe the methodology that they used in testing the two real-world EVMs and the security problems they found and stated the learning of their analysis, and the improvements needed. In this paper, author describe the methodology that was used in testing the two real-world electronic voting systems we evaluated, the findings of our analysis, our attacks, and the lessons we learned.

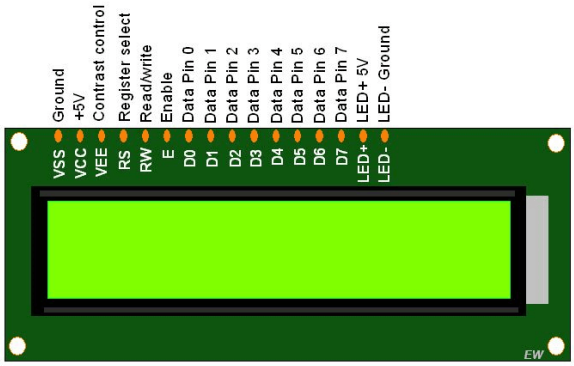
III. METHODOLOGY

A. Microcontroller



The microcontroller sends the signal given from switches and decides the mode of operation. In voting mode, it increments the data for the corresponding key i.e., respective candidate as well as it sends signal to display block to indicate one key is pressed. In counting mode, microcontrollers fetch data from memory location and send it to display devices.

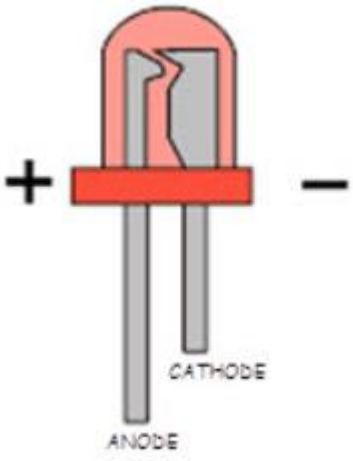
B. LCD



Liquid Crystal Display which is commonly known as LCD is an alphanumeric display it means that it can display alphabets, numbers as well as special symbols thus LCD is a user- friendly display device which can be used for displaying various messages unlike seven segment display which can display only number and some of the alphabets. The only disadvantages of LCD over seven segments are that seven segment is robust display and can be visualised from a longer distance as compared to LCD. Here I

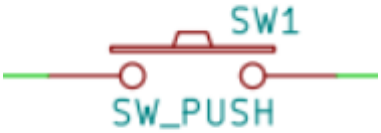
have used 16*2 alphanumeric display which means on this display I can display two lines with maximum of 16 character in one line.

C. *LED*



Light Emitting Diode is a semiconductor light source. LEDs are used as indicator lamps in many devices and are increasingly used for other lighting. LEDs are used in application as driver as replacement for aviation lighting, automotive lighting (particularly brake lamps, turn signal and indicators) as well as in traffic signal.

D. *CONTROL SWITCHES*



There are 4 control switches: Start Switch, Check Switch, Authority Switch, Result Switch

Voting Mode: When toggle switch is in voting mode "Voting mode" is displayed followed by "Please vote". After a vote being given, "Please wait for authority switch" is displayed and again enable for voting after Control switch being pressed by the voting Authority.

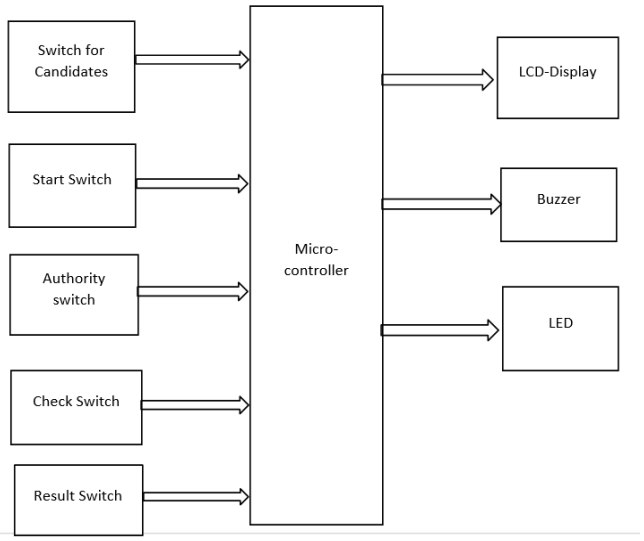
Counting Mode: When toggle switch is in counting mode "Counting mode" in displayed on the screen, and total number of votes to respective candidate can be displayed on the screen by pressing the respective key assigned to them.

Check mode: Press check switch to check the voting status in between the voting process.

Buzzer indication: Pressing of key in voting mode is indicated by a buzzer sound.

Controller switch: This switch is provided for enabling the next entery in voting mode. This switch is under the control of voting authority.

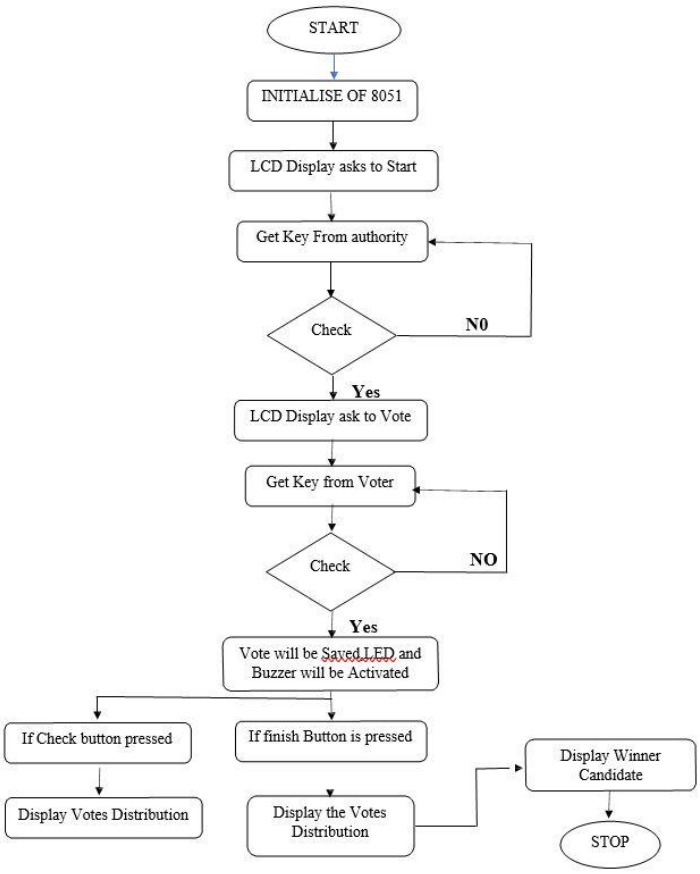
IV. BLOCK DIAGRAM OF ELECTRONIC VOTING MACHINE



V. SOFTWARE FLOW CHART

KEIL SOFTWARE*: The Keil 8051 Development Tools are designed to solve the complex problems facing embedded software developers.*

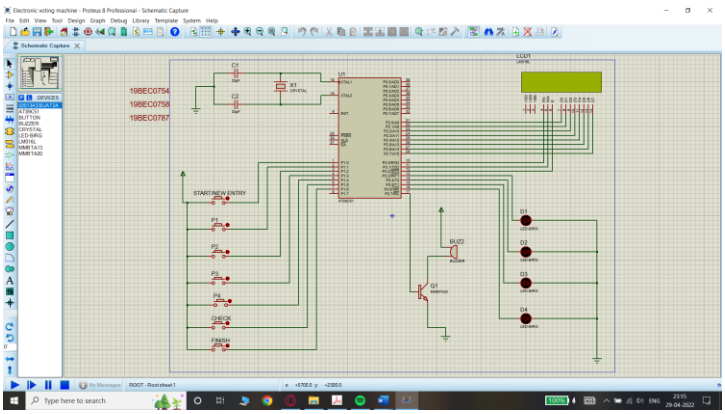
PROTEUS SOFTWARE*: Proteus 8.0 is a Virtual System Modelling (VSM) that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs. This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and/or LED and LCD displays and, if attached to the PC, switches and buttons.*



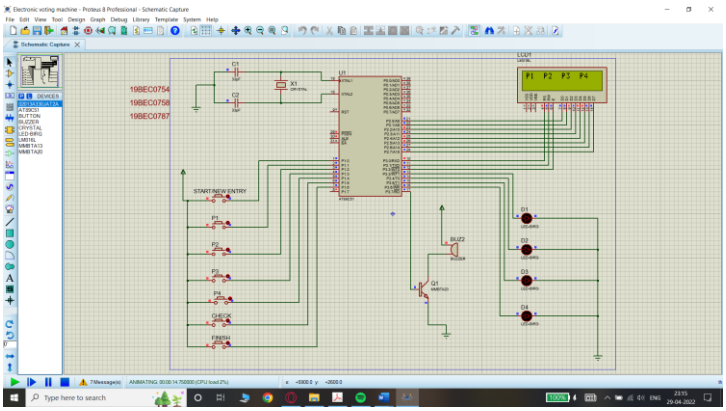
VI. RESULTS

We used the microcontroller to build an electronic voting machine and successfully got the poll and polling results.

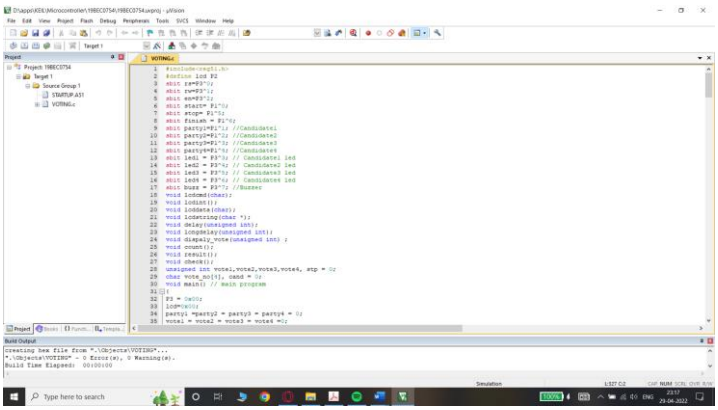
A. *Circuit Diagram*



B. *Simulating circuit*



C. *Keil simulator screenshot(the code is too big)*



VII. CONCLUSION AND FUTURE WORK

In this project, we have described the specification and architecture of an electronic voting machine.Various fault-tolerance and security issues are delegated to the platform itself, therefore relieving the application designer from accommodating these features in the application design itself. This approach allows for the easy development and deployment of applications. For quite some time, voting equipment vendors have maintained that their systems are secure, and that the closed-source nature makes them even more secure. Our glimpse into the code of such a system reveals that there is little difference in the way code is developed for voting machines relative to other commercial endeavours. In fact, we believe that an open process would result in more careful development, as more scientists, software engineers, political activists, and others who value their democracy would be paying attention to the quality of the software that is used for their elections. (Of course, open source would not solve all of the problems with electronic elections. It is still important to verify somehow that the binary program images running in

the machine correspond to the source code and that the compilers used on the source code are non-malicious. However, open source is a good start.) Such open design processes have proven successful in projects ranging from very focused efforts, such as specifying the Advanced Encryption Standard (AES), through very large and complex systems such as maintaining the Linux operating System. Australia is currently using an open source voting system10Alternatively, security models such as the voter-verified audit trail allow for electronic voting systems that produce a paper trail that can be seen and verified by a voter. In such a system, the correctness burden on the voting terminal's code is significantly less as voters can see and verify a physical object that describes their vote. Even if, for whatever reason, the machines cannot name the winner of an election, then the paper ballots can be recounted, either mechanically or manually, to gain progressively more accurate election results. Voter-verifiable audit trails are required in some U.S. States, and major DRE vendors have made public statements that they would support such features if their customers required it. The EVM project an ambitious attempt to create an open-source voting system with a voter-verifiable audit trail a laudable goal The model where individual vendors write proprietary code to run our elections appears to be unreliable, and if we do not change the process of designing our voting systems, we will have no confidence that our election results will reflect the will of the electorate. We owe it to ourselves and to our future to have robust, well-designed election systems to preserve the bedrock of our democracy.

Finally, we can also add another security like a keypad to make the voting machine more secure which will only allow the voting machine to be accessed only when the code is entered in the keypad

VIII. REFERENCES

[1]S. K. Shaw, S. Poddar, V. Singh and S. Dogra, "Design and Implementation of Arduino Based Voting Machine," 2018 IEEE Electron Devices Kolkata Conference (EDKCON), 2018, pp. 450-454, doi: 10.1109/EDKCON.2018.8770474.
https://ieeexplore.ieee.org/document/8770474

[2]Arduino based smart electronic voting machine, V. K. Priya, V. Vimaladevi, B. Pandimeenal and T. Dhivya, 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017, pp. 641-644, doi: 10.1109/ICOEI.2017.8300781.
https://ieeexplore.ieee.org/document/8300781

[3]YashalDorzeeLepcha, Deepak Rasaily, Jigmee Sherpa"Design of Electronic Voting Machine using Microcontroller", International Journal of Engineering Trends and Technology (IJETT), V32(5),225-227 February 2016. ISSN:2231-5381.
http://ijettjournal.org/archive/ijett-v32p246

[4]S. M. Ali et al., "Micro-controller based smart electronic voting machine system," IEEE International Conference on Electro/Information Technology, 2014, pp. 438-442, doi: 10.1109/EIT.2014.6871804.
https://ieeexplore.ieee.org/document/6871804

[5]D. Balzarotti et al., "An Experience in Testing the Security of Real-World Electronic Voting Systems," in IEEE Transactions on Software Engineering, vol. 36, no. 4, pp. 453-473, July-Aug. 2010, doi: 10.1109/TSE.2009.53.
https://ieeexplore.ieee.org/document/5210119

CODE

```
#include<reg51.h>
#define lcd P2
sbit rs=P3^0;
sbit rw=P3^1;
sbit en=P3^2;
```

```
sbit start= P1^0;
sbit stop= P1^5;
sbit finish = P1^6;
sbit party1=P1^1; //Candidate1
sbit party2=P1^2; //Candidate2
sbit party3=P1^3; //Candidate3
sbit party4=P1^4; //Candidate4
sbit led1 = P3^3; // Candidate1 led
sbit led2 = P3^4; // Candidate2 led
sbit led3 = P3^5; // Candidate3 led
sbit led4 = P3^6; // Candidate4 led
sbit buzz = P3^7; //Buzzer
void lcdcmd(char);
void lcdint();
void lcddata(char);
void lcdstring(char *);
void delay(unsigned int);
void longdelay(unsigned int);
void dispaly_vote(unsigned int) ;
void count();
void result();
void check();
unsigned int vote1,vote2,vote3,vote4, stp = 0;
char vote_no[4], cand = 0;
void main() // main program
{
P3 = 0x00;
lcd=0x00;
party1 =party2 = party3 = party4 = 0;
vote1 = vote2 = vote3 = vote4 =0;
start = stop = finish = 0;
lcdint();
top:
lcdcmd(0x81);
lcdstring("Press start to");
lcdcmd(0xc3);
lcdstring("initiate");
while(1)
{
        if(start==1)
{
lcdcmd(0x01);
lcdcmd(0x84);
lcdstring("WELCOME!!");
longdelay(200);
lcdcmd(0x01);
lcdcmd(0x81);
lcdstring("CAST YOUR VOTE");
longdelay(100);
lcdcmd(0x01);
lcdstring("P1");
delay(500);
lcdcmd(0x84);
lcdstring("P2");
delay(500);
lcdcmd(0x88);
lcdstring("P3");
delay(500);
lcdcmd(0x8C);
lcdstring("P4");
count();
lcdcmd(0x01);
```

```
lcdcmd(0x80);
lcdstring("Voted candidate");
lcdcmd(0xc5);
if(cand == 1)
lcdstring("P1");
if(cand == 2)
lcdstring("P2");
if(cand == 3)
lcdstring("P3");
if(cand == 4)
lcdstring("P4");
longdelay(200);
lcdcmd(0x01);
lcdcmd(0x83);
lcdstring("Thank you!");
longdelay(300);
lcdcmd(0x01);
goto top;
}
if(stop == 1)
{
stp = 1;
            }
if(stp == 1)
{
check();
stp = 0;
}
if(finish == 1)
{
while(1)
{
result();
}
}
}
}
void check() // check count
{
lcdcmd(0x01);
lcdstring("P1");
delay(500);
lcdcmd(0x84);
lcdstring("P2");
delay(500);
lcdcmd(0x88);
lcdstring("P3");
delay(500);
lcdcmd(0x8C);
lcdstring("P4");
lcdcmd(0xc0);
dispaly_vote(vote1);
lcdcmd(0xc4);
dispaly_vote(vote2);
lcdcmd(0xc8);
dispaly_vote(vote3);
lcdcmd(0xcc);
dispaly_vote(vote4);
longdelay(300);
}
void result() // Display result
{
```

```
            int max=0,flag=0;
lcdcmd(0x01);
lcdstring("P1");
delay(500);
lcdcmd(0x84);
lcdstring("P2");
delay(500);
lcdcmd(0x88);
lcdstring("P3");
delay(500);
lcdcmd(0x8C);
lcdstring("P4");
lcdcmd(0xc0);
dispaly_vote(vote1);
lcdcmd(0xc4);
dispaly_vote(vote2);
lcdcmd(0xc8);
dispaly_vote(vote3);
lcdcmd(0xcc);
dispaly_vote(vote4);
if(vote1>max)
{
max=vote1;
}
if(vote2>max)
{
max=vote2;
}
if(vote3>max)
{
max=vote3;
}
if(vote4>max)
{
max=vote4;
}
longdelay(400);
if ( (vote1 == max) && ( vote2 != max) && (vote3 != max)&& (vote4 != max) )
{
flag = 1;
lcdcmd(0x01);
lcdcmd(0x80);
lcdstring("P1");
lcdcmd(0xc5);
lcdstring("WINS");
longdelay(400);
            }
if ( (vote2 == max) && ( vote1 != max) && (vote3 != max)&& (vote4 != max) )
{
flag = 1;
lcdcmd(0x01);
lcdcmd(0x80);
lcdstring("P2");
lcdcmd(0xc5);
lcdstring("WINS");
longdelay(400);
}
if ( (vote3 == max) && ( vote2 != max) && (vote1 != max)&& (vote4 != max) )
{
flag = 1;
lcdcmd(0x01);
lcdcmd(0x80);
```

```
lcdstring("P3");
lcdcmd(0xc5);
lcdstring("WINS");
longdelay(400);
}
if ( (vote4 == max) && ( vote2 != max) && (vote1 != max)&& (vote3!= max) )
{
flag = 1;
lcdcmd(0x01);
lcdcmd(0x80);
lcdstring("P4");
lcdcmd(0xc5);
lcdstring("WINS");
longdelay(400);
}
if(flag==0)
{
lcdcmd(0x01);
lcdcmd(0x80);
lcdstring("clash between");
lcdcmd(0xc0);
if(vote1==max)
{
lcdstring("P1 ");
}
if(vote2==max)
{
lcdstring("P2 ");
}
if(vote3==max)
            {
lcdstring("P3 ");
}
if(vote4==max)
{
lcdstring("P4 ");
}
longdelay(100);
}
}
void dispaly_vote(unsigned int vote) // send 0-9 character values (ASCII)
{
int k,p;
for (k=0; k<=2; k++)
{
vote_no[k]=vote%10;
vote=vote/10;
}
for (p=2; p>=0; p--)
{
lcddata(vote_no[p]+0x30);
}
}
void count() // count votes
{
while(party1==0&&party2==0&&party3==0&&party4==0);
if (party1==1)
{
vote1 = vote1 + 1;
cand = 1;
led1 = 1;
buzz = 1;
```

```
longdelay(100);
buzz = 0;
led1 = 0;
}
if (party2==1)
{
vote2 = vote2 + 1;
cand = 2;
led2 = 1;
            buzz = 1;
longdelay(100);
buzz = 0;
led2 = 0;
}
if (party3==1)
{
vote3 = vote3 + 1;
cand = 3;
led3 = 1;
buzz = 1;
longdelay(100);
buzz = 0;
led3 = 0;
}
if (party4==1)
{
vote4 = vote4 + 1;
cand = 4;
led4 = 1;
buzz = 1;
longdelay(100);
buzz = 0;
led4 = 0;
}
}
// lcd programming
void delay(unsigned int x)
{
unsigned int i;
for(i=0; i<x; i++);
}
void longdelay(unsigned int u)
{
unsigned int i,j;
for(i=0; i<u; i++)
for(j=0; j<1275; j++);
}
void lcdint()
{
lcdcmd(0x38);
lcdcmd(0x01);
lcdcmd(0x0c);
            lcdcmd(0x80);
}
void lcdcmd(char value)
{
lcd = value;
rw=0;
rs=0;
en=1;
delay(500);
en=0;
```

```
}
void lcdstring(char *p)
{
while(*p!='\0')
{
lcddata(*p);
delay(2000);
p++;
}
}
void lcddata(char value)
{
lcd = value;
rs=1;
rw=0;
en=1;
delay(500);
en=0;
}
```