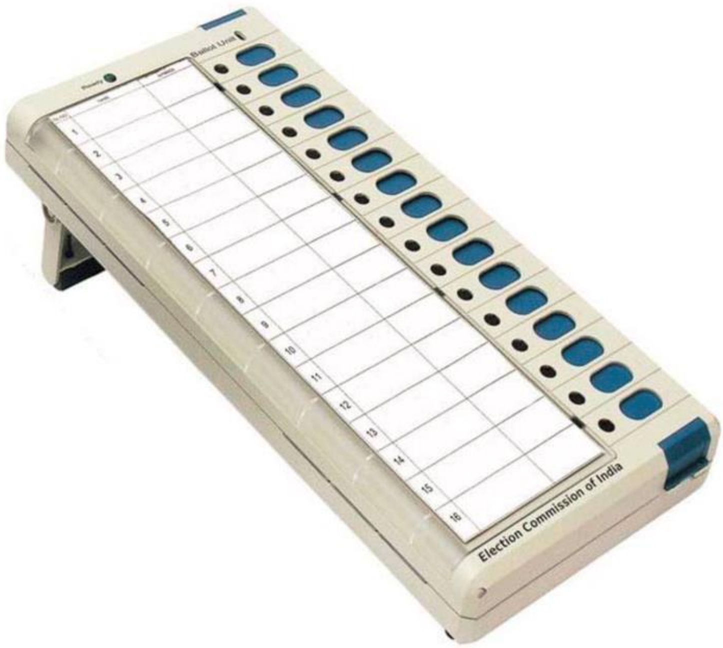
# 1. INTRODUCTION

In today’s fast changing world, Electronic Voting Machine (EVM) retains all the characteristics of voting by ballot papers, while making polling a lot more expedient. Being fast and absolutely reliable, the EVM saves considerable time, money and manpower. And, of course, helps maintain total voting secrecy without the use of ballot papers. The EVM is 100 per cent tamper proof. And, at the end of the polling, just press a button and there you have the results.




The EVMs were devised and designed by Election Commission of India in collaboration with two Public Sector undertakings viz., Bharat Electronics Limited, Bangalore and Electronics Corporation of India Limited, Hyderabad. In 1980, Mr. M. B. Haneefa designed the first Indian voting machine, gazetted “Electronically operated vote counting machine” (Gazette: 191/Mas/80, 15 October 1980). His original design was exhibited to the public in six Government Exhibitions of Tamil Nadu, viz.: Madras (Chennai), Trichy, Coimbatore, Salem, Madurai and Tirunelveli. Remarkably, the voting machine was designed using only Integrated Circuits. EVMs were first used in 1981 in the by-election to North Paravur Assembly Constituency of Kerala for a limited number of polling stations (50 polling stations).

# 2. EQUIPMENT AND PCB DESIGN

## 2.1 Equipment:

The different major equipment used in system are enlisted below:

a) Microcontroller 8051
b) LCD
c) Push Buttons
d) Power supply
e) LED
f) PCB Board

### 2.1.1 Microcontroller 8051:

A microcontroller (also MCU or µC) is a functional computer system-on-a-chip. It contains a processor core, memory, and programmable input/output peripherals. Microcontrollers include an integrated CPU, memory (a small amount of RAM, program memory, or both) and peripherals capable of input and output. It emphasizes high integration, in contrast to a microprocessor which only contains a CPU (the kind used in a PC). In addition to the usual arithmetic and logic elements of a general purpose microprocessor, the microcontroller integrates additional elements such as read-write memory for data storage, read-only memory for program storage, Flash memory for permanent data storage, peripherals, and input/output interfaces. At clock speeds of as little as 32 KHz, microcontrollers often operate at very low speed compared to microprocessors, but this is adequate for typical applications. They consume relatively little power (milli watts), and will generally have the ability to retain functionality while waiting for an event such as a button press or interrupt. Power consumption while sleeping (CPU and peripherals off) may be just Nano watts, making them ideal for low power and long lasting battery applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, remote controls, office machines, appliances, power tools, and toys. By reducing the size, cost, and power consumption compared to a design using a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to electronically control many more processes.

The **AT89S52** is a low-power, high-performance CMOS 8-bit microcontroller with 8Kbytes of in-system programmable Flash memory.

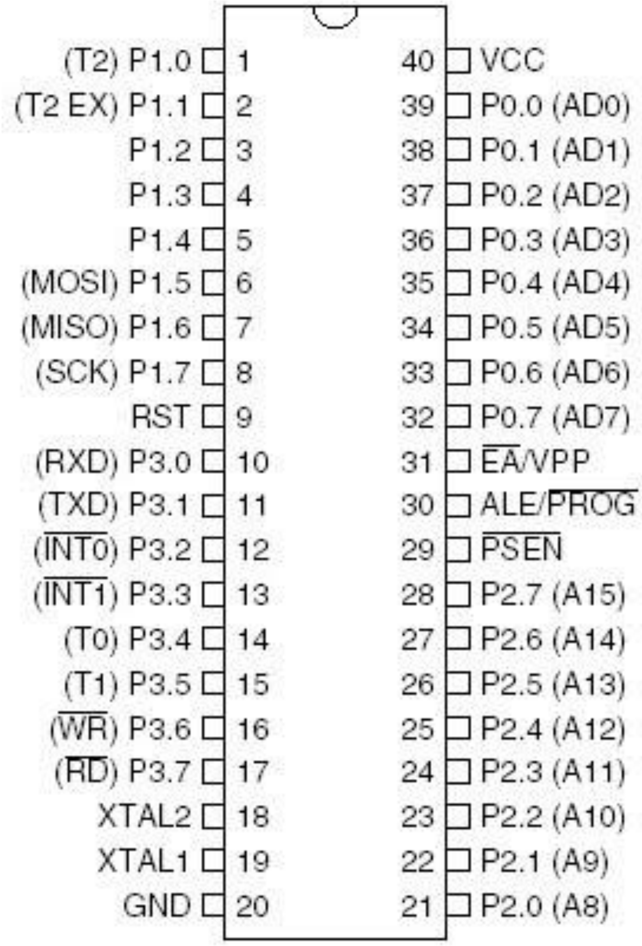



**Fig.2.1.1.showing AT89S52**

## Pin Description :

**Vcc :** Power **supply**

**Gnd :** Ground

**Port 0**: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

**Port 1**: Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table. Port 1 also receives the low-order address bytes during Flash programming and verification.

| Port Pin | Alternate Functions |
|---|---|
| P1.0 | T2 (external count input to Timer/Counter 2), clock-out |
| P1.1 | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| P1.5 | MOSI (used for In-System Programming) |
| P1.6 | MISO (used for In-System Programming) |
| P1.7 | SCK (used for In-System Programming) |

**Table2.1.1.**showing functions of port 1 sub pins

**Port 2** : Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

**Port 3**: Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups. Port 3 receives some control signals for Flash programming and verification. Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

| Port Pin | Alternate Functions |
|---|---|
| P3.0 | RXD (serial input port) |
| P3.1 | TXD (serial output port) |
| P3.2 | $\overline{\text{INT0}}$ (external interrupt 0) |
| P3.3 | $\overline{\text{INT1}}$ (external interrupt 1) |
| P3.4 | T0 (timer 0 external input) |
| P3.5 | T1 (timer 1 external input) |
| P3.6 | $\overline{\text{WR}}$ (external data memory write strobe) |
| P3.7 | $\overline{\text{RD}}$ (external data memory read strobe) |

**Table 2.1.2**showing functions of sub pins of port 3

**RST:** Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

**ALE/PROG**: Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

**PSEN**:  Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

**EA/VPP**: External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

**XTAL1**: Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2:** Output from the inverting oscillator amplifier.

### 2.1.2 LCD:

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively

This LCD's typically have 14 data pins and 2 for the LED backlight. Character LCDs use a standard 14-pin interface and those with backlights have 16 pins. There is also be a single backlight pin, with the other connection via Ground or VCC pin. The two backlight pins may precede the pin 1. The nominal backlight voltage is around 4.2V at 25˚C using a VDD 5V capable model.

#### 2.1.2.1 Features:

a) 5 ´ 8 and 5 ´ 10 dot matrix possible

b) Low power operation support: 2.7 to 5.5V

c) Wide range of liquid crystal display driver power 3.0 to 11V

d) 80 ´ 8-bit display RAM (80 characters max.)

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

| POSITION | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | LINE1 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |

**Fig.2.1.2.**showing address positions of LCD display

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exists. “super twist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

**PIN DESCRIPTION:**

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

**Fig2.1.3:** pin diagram of 16X2 lines LCD

| PIN | SYMBOL | FUNCTION |
|---|---|---|
| 1 | Vss | Power Supply(GND) |
| 2 | Vdd | Power Supply(+5V) |
| 3 | Vo | Contrast Adjust |
| 4 | RS | Instruction/Data Register Select |
| 5 | R/W | Data Bus Line |
| 6 | E | Enable Signal |
| 7-14 | DB0-DB7 | Data Bus Line |
| 15 | A | Power Supply for LED B/L(+) |
| 16 | K | Power Supply for LED B/L(-) |

**Table.2.1.3**.showing pin functions of LCD

## 2.1.3 POWER SUPPLY:

Power supply is a reference to a source of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others

This power supply section is required to convert AC signal to DC signal and also to reduce the amplitude of the signal. The available voltage signal from the mains is 230V/50Hz which is an AC voltage, but the required is DC voltage(no frequency) with the amplitude of +5V and +12V for various applications.

In this section we have Transformer, Bridge rectifier, are connected serially and voltage regulators for +5V and +12V (7805 and 7812) via a capacitor (1000µF) in parallel are connected parallel as shown in the circuit diagram below. Each voltage regulator output is again is connected to the capacitors of values (100µF, 10µF, 1 µF, 0.1 µF) are connected parallel through which the corresponding output(+5V or +12V) are taken into consideration.

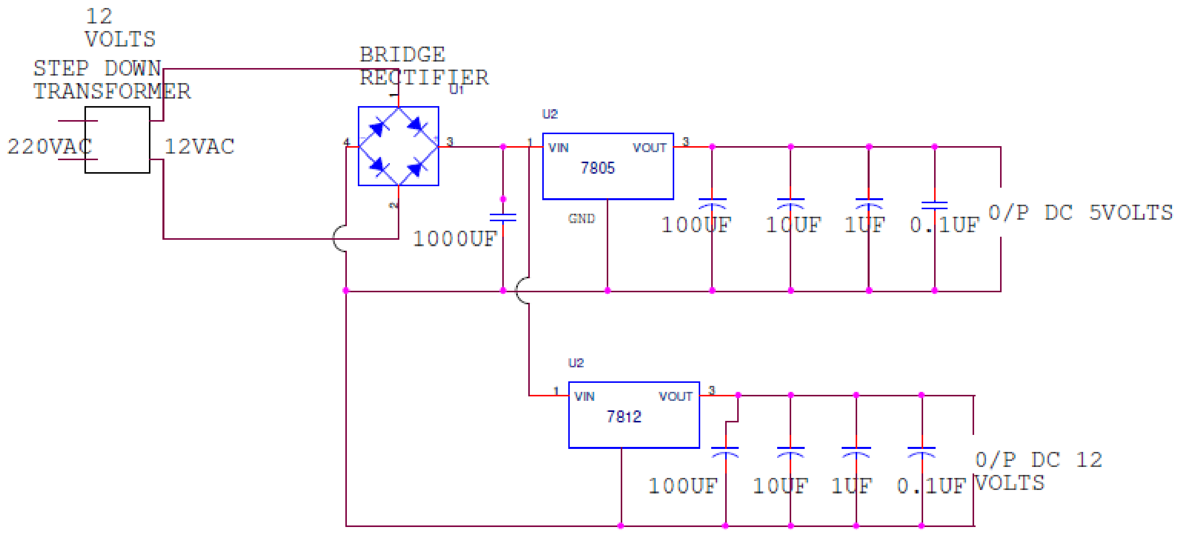
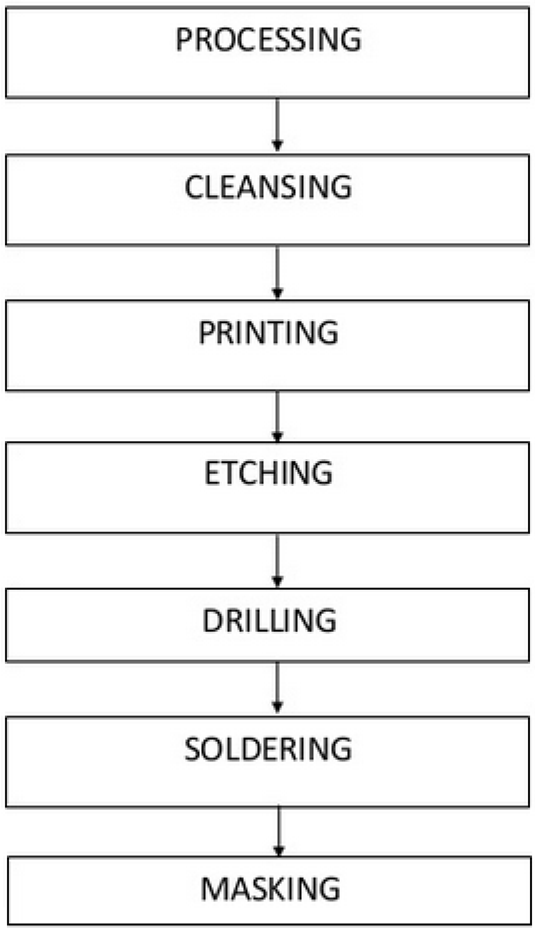



**Fig.2.1.4.** Diagram showing Power supply circuit for project

## 2.2 PCB (PRINTED CIRCUIT BOARD) DESIGNING:

A printed circuit board, or PCB, is used to mechanically support and electrically connect electronic components using conductive pathways, tracks or traces etched from copper sheets laminated onto a non-conductive substrate. It is also referred to as printed wiring board (PWB) or etched wiring board. A PCB populated with electronic components is a printed circuit assembly (PCA), also known as a printed circuit board assembly (PCBA) PCBs are inexpensive, and can be highly reliable. They require much more layout effort and higher initial cost than either wire-wrapped or point-to-point constructed circuits, but are much cheaper and faster for high-volume production. Much of the electronics industry's PCB design, assembly, and quality control needs are set by standards. PCB design includes the following steps:


PROCESSING
CLEANSING
PRINTING
ETCHING
DRILLING
SOLDERING
MASKING

# 3. KEIL COMPILER (SOFTWARE REQUIREMENT)

The Keil C51 C Compiler for the 8051 microcontroller is the most popular 8051 C compiler in the world. It provides more features than any other 8051 C compiler available today.

The C51 Compiler allows you to write 8051 microcontroller applications in C that, once compiled, have the efficiency and speed of assembly language. Language extensions in the C51 Compiler give you full access to all resources of the 8051.

The C51 Compiler translates C source files into relocatable object modules which contain full symbolic information for debugging with the µVision Debugger or an in-circuit emulator. In addition to the object file, the compiler generates a listing file which may optionally include symbol table and cross reference information.

## Features :

- Nine basic data types, including 32-bit IEEE floating-point,
- Flexible variable allocation with **bit**, **data**, **bdata**, **idata**, **xdata**, and **pdata** memory types,
- Interrupt functions may be written in C,
- Full use of the 8051 register banks,
- Complete symbol and type information for source-level debugging,
- Use of **AJMP** and **ACALL** instructions,
- Bit-addressable data objects,
- Built-in interface for the RTX51 Real Time Kernal
- Support for dual data pointers on Atmel, AMD, Cypress, Dallas Semiconductor, Infineon, Philips, and Triscend microcontrollers,
- Support for the Philips 8xC750, 8xC751, and 8xC752 limited instruction sets,
- Support for the Infineon 80C517 arithmetic unit.

# 4. WORKING OPERATION & IMPLEMENTATION

This chapter deals with working of Electronic Voting Machine. This chapter also introduces programming used in microcontroller.

## 4.1 WORKING:

Operation or Working of Project:

1. Power on: When supply is turned on RED LED glows.

2. Mode selection:

i. Voting mode: toggle switch on VCC

ii. Counting mode: toggle switch on GND.

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.

3. Clear mode: Press clear switch when all entries are required to be erased. Clear switch should be pressed before voting procedure.

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

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

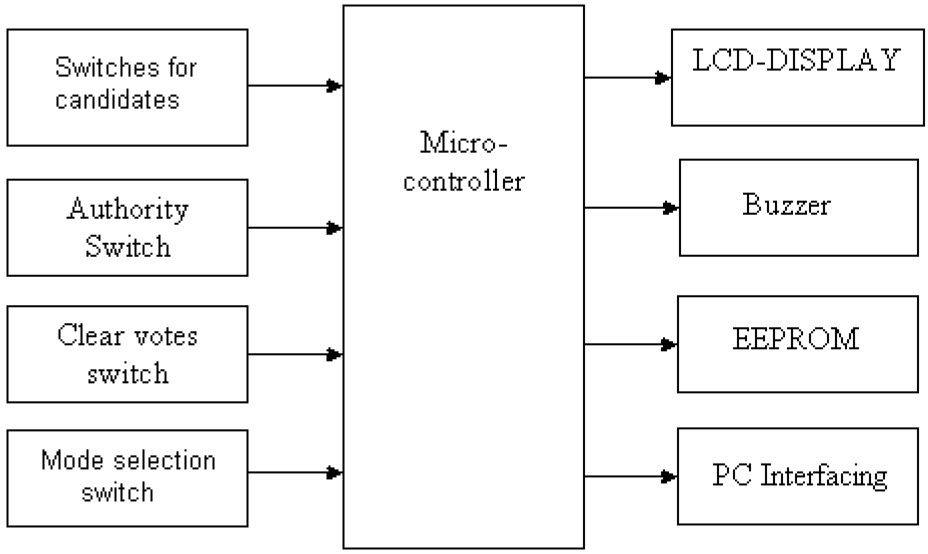



**Fig.4.1.1**.Showing Block diagram of GSM based home device controlling system

## 4.2 CIRCUIT DIAGRAM:

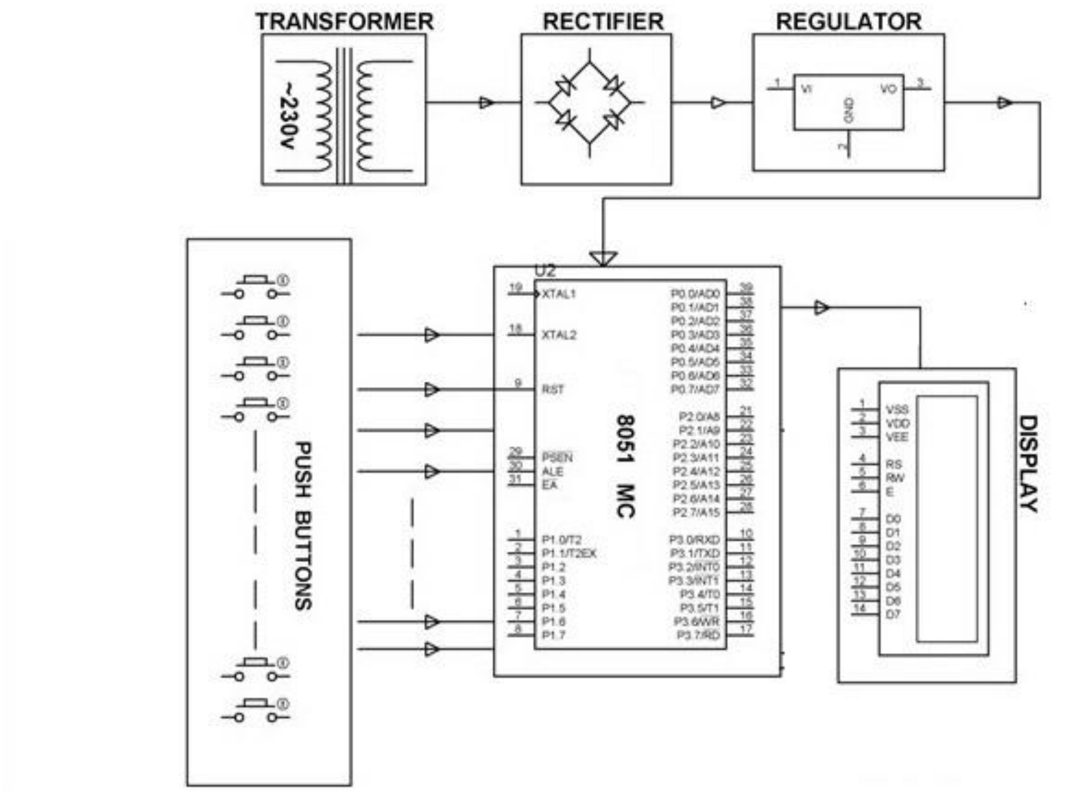



**Fig.4.2.1.**Showing circuit diagram of GSM based home device controlling system

## 4.3 HARDWARE IMPLEMENTATION :

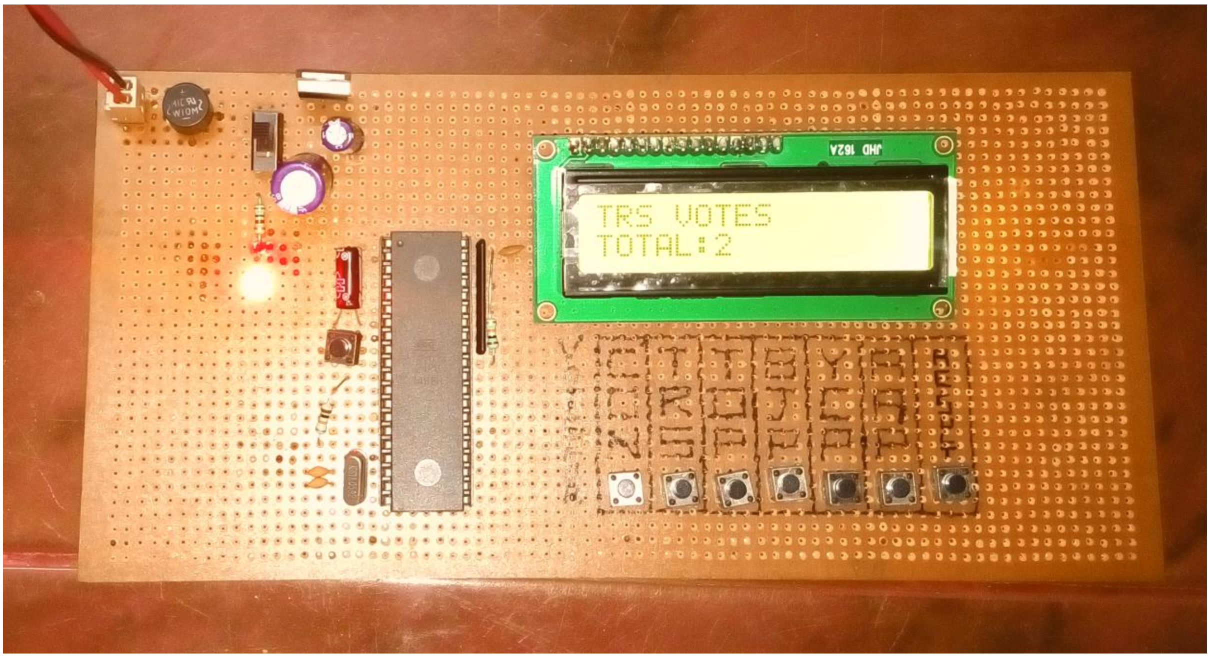



**Fig.4.3.1**.Showing hardware implementation of Project

# 5.ADVANTAGES AND DISADVANTAGES

## 5.1 ADVANTAGES:

1. Man power required for counting of votes minimal.

2. Eliminates the cost of printing ballot papers which is nearly 30% of election budget

3. Can be easily transported.

4. Eliminates the menace of "Invalid Votes"

5. Results are obtained with the press of a button.

6. Administrative expenses reduced.

## 5.2 DISADVANTAGES:

1. Inequality problem because for those constituencies with low income/investment might not able to afford the equipment for electronic voting

2. Secondly, vulnerable to security the security issue is the main drawback of the electronic voting system. The attacks might be happened from the webpage, network, to the extent of server's database. Virus The server can be easily protected from the attack of virus by using some specific kinds of operating system. Data will be dangerous if virus attack.

# 6. APPLICATIONS

The previous chapter deal with stated merits and de-merits of the project, the project can be put for use in following fields-

1. It can be used in school, college student union elections.

2. It can be used to find the general opinion of people on various issues, anywhere where majority opinion is to be found out.

# APPENDIX : Source Code

```
#include<reg51.h>

sfr input=0x90;

sfr ldata=0xa0;

sbit rs=P0^7;

sbit rw=P0^6;

sbit en=P0^5;

sbit m=P3^0;

sbit n=P3^1;

sbit buzz=P3^2;

sbit on=P3^3;

void delay(int );

void lcdcmd(char );

void lcddata1(char *);

void lcddata(char);

int i1,i11,i12,i2,i21,i22,i3,i31,i32,i4,i41,i42,i5,i51,i52,i6,i61,i62,i7,i71,i72,i8,i81,i82=0;

void main()

{

on=0;

P1=0;P3=0;

while(1)

{

lcdcmd(0x38);
```

```
delay(10);

lcdcmd(0x0e)   ;

delay(10);

lcdcmd(0x01);

lcdcmd(0x06)  ;

delay(20) ;

if(n==1)

on=1;

if(m==1&on==1)

{

if (input==0x01)

{

buzz=1;

while (input == 0x01);

i1=i1 + 1;

if(i1>=10)

i11=i1/10;

i12=i1%10;

on=0;

buzz=0;

}

if (input==0x02)

{buzz=1;
```

```
while (input == 0x02);

{

i2=i2 + 1;

if(i2>=10)

i21=i2/10;

i22=i2%10;

on=0;

buzz=0;

}

}

if (input==0x04)

{

buzz=1;

while (input ==0x04);

{

i3=i3 + 1;

if(i3>=10)

i31=i3/10;

i32=i3%10;

on=0;

buzz=0;

}

}
```

```
if (input==0x08)

{

buzz=1;

while (input == 0x08);

{

i4=i4 + 1;

if(i4>=10)

i41=i4/10;

i42=i4%10;

on=0;

buzz=0;

}

}

if (input==0x10)

{

buzz=1;

while (input == 0x10);

{

i5=i5 + 1;

if(i5>=10)

i51=i5/10;

i52=i5%10;

i5=0;
```

```
on=0;

buzz=0;

}

}

if (input==0x20)

{

buzz=1;

while (input == 0x20);

{

i6=i6 + 1;

if(i6>=10)

i61=i6/10;

i62=i6%10;

on=0;

buzz=0;

}

}

if (input==0x40)

{

buzz=1;

while (input == 0x40);

{

i7=i7 + 1;
```

```
if(i7>=10)

i71=i7/10;

i72=i7%10;

on=0;

buzz=0;

}

}

if (input==0x80)

{

buzz=1;

while (input == 0x80);

{

i8=i8 + 1;

if(i7>=10)

i81=i8/10;

i82=i8%10;

on=0;

buzz=0;

}

}

}

 if(m==0)

{
```

```
if (input==0x01)

{

lcddata1(" CONGRESS=");

lcddata(i11+0x30);

lcddata(i12+0x30);

delay(100);

}

if (input==0x02)

{

lcdcmd(0x01);

lcddata1(" BJP=");

lcddata(i21+0x30);

lcddata(i22+0x30);

delay(100);

}

if (input==0x04)

{

lcdcmd(0x01);

lcddata1(" CP(M&I)=");

lcddata(i31+0x30);

lcddata(i32+0x30);

delay(100);

}
```

```
if (input==0x08)

{

lcdcmd(0x01);

lcddata1(" TDP=");

lcddata(i41+0x30);

lcddata(i42+0x30);

delay(100);

}

if(input==0x10)

{

lcdcmd(0x10);

lcddata1(" TRS=");

lcddata(i51+0x30);

lcddata(i52+0x30);

delay(100);

}

if(input==0x20)

{

lcdcmd(0x01);

lcddata1(" PRP=");

lcddata(i61+0x30);

lcddata(i62+0x30);

delay(100);
```

```
}
if(input==0x40)
{
lcdcmd(0x01);
lcddata1(" INDEPENDANT1=");
lcddata(i71+0x30);
lcddata(i72+0x30);
delay(100);
}
if(input==0x80)
{
lcdcmd(0x01);
lcddata1(" INDEPENDANT2=");
lcddata(i81+0x30);
lcddata(i82+0x30);
delay(100);
}
Else
{
lcdcmd(0x01);
lcddata1("press key  ");
delay(100) ;
}
```

```
}
}
}
void delay(int time)
{
int i,j;
for(i=0;i<time;i++)
for(j=0;j<900;j++);
}
void lcdcmd(char value)
{
ldata=value;
rs=0;
rw=0;
en=1;
delay(2);
en=0;
}
void lcddata1(char *value)
{
int i;
for(i=0;value[i]!='\0';i++)
{
```

```
ldata=value[i];

rs=1;

rw=0;

en=1;

delay(1);

en=0;

}

}

void lcddata(char value)

{

ldata=value;

rs=1;

rw=0;

en=1;

delay(1);

en=0;

}
```

# REFERENCES :


- http://www.dnatechindia.com/Projects/Synopsis-and-Abstract/Electronic-Voting-Machine.html
- https://en.wikipedia.org/wiki/Electronic_voting
- http://www.dnatechindia.com/Projects/Synopsis-and-Abstract/Electronic-Voting-Machine.html
- http://www.projectsof8051.com/microcontroller-based-electronic-voting-machine/
- http://livelyviews.com/what-is-evms-i-e-electronic-voting-machines/