



**SMART ELECTRONIC VOTING
MACHINE
PROJECT REPORT**



Submitted by

BHARANIS.C.

Reg. No.:13BEC025

in partial fulfillment for the award of the degree

of

**BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION
ENGINEERING**

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2017

**KUMARAGURU COLLEGE OF
TECHNOLOGY COIMBATORE – 641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report titled “**SMART ELECTRONIC VOTING MACHINE**” is the bonafide work of “**BHARANI.S.C. (13BEC025)**” who carried out the project work under my supervision.

SIGNATURE

Mr.S.Govindaraju,M.E.,
PROFESSOR
Department of ECE
Kumaraguru College of Technology
Coimbatore-641049

SIGNATURE

Dr. K.Malarvizhi M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Department of ECE
Kumaraguru College of Technology
Coimbatore- 641049

The candidate with Register number 13BEC025 is examined by us in the project viva-voce examination held on 21/04/2017.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First I would like to express our praise and gratitude to the Lord, who has showered his grace and blessing enabling me to complete this project in an excellent manner. He has made all things in beautiful in his time.

I express my sincere thanks to our beloved Joint Correspondent, **Shri. Shankar Vanavarayar** for his kind support and for providing necessary facilities to carry out the project work.

I would like to express my sincere thanks to our beloved Principal **Dr.R.S.Kumar M.E., Ph.D.**, who encouraged me with his valuable thoughts.

I would like to express my sincere thanks and deep sense of gratitude to our HOD, **Dr.K.Malarvizhi, M.E.,Ph.D.**, for her valuable suggestions and encouragement which paved way for the successful completion of the project.

I am greatly privileged to express my deep sense of gratitude to the Project Coordinator and guide **Mr.S.Govindaraju M.E.**, Professor, for his continuous support and expert counselling throughout the course.

I wish to convey my deep sense of gratitude to all teaching and non-teaching staff members of ECE Department for their help and cooperation.

Finally, I thank my parents and my family members for giving me the moral support in all of my activities and my dear friends who helped me to endure my difficult times with their unfailing support and warm wishes.

ABSTRACT

The main objective of this project is to provide instant Voting, and to reduce the manpower involved in voter authentication, voting and ballot security. Authentication of the voter is done by OTP technique. OTP(One Time Password) is given to the voter on the day of election by the Admin at the main server with the aadhar number being used as the user name (as used in this system). Ballot units which are not the usual ones are kept at different locations (users handle this) which consist of a controller, LCD display and keypad matrix. Once the voter enters his/her aadhar number from the mobile (GSM module) it will be received at the server end. An OTP will be sent to the user from the server unit. Now the voter has to enter the OTP such that the list of parties contesting in election will be displayed instantaneously in the LCD unit. Now the voter is left free to cast the vote. The casted votes are received at the main server, the time the voter casts the vote. Voter details are kept confidential such that no access could be made to know who the voter is during the counting of votes.

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE No
	ABSTRACT	4
1	INTRODUCTION	7
2	BLOCK DIAGRAM	8
3	COMPONENTS	8
	3.1 PIC 16F877A	9
	3.1a PIN CONFIGURATION	10
	3.1b FEATURES	11
	3.2 LCD DISPLAY	12
	3.2a PIN DETAILS	13
	3.3 KEYPAD MATRIX	14
	3.3a FEATURES	15
	3.4 GSM MODULE	15
	3.5 POWER SUPPLY	16
	3.6 RS-232	17
4	MPLAB IDE	18
5	ALGORITHM	19
6	FLOWCHART	20
7	WORKING	21
8	RESULT AND CONCLUSION	22
9	FUTURE SCOPE	22
10	PROGRAM CODING	23
11	REFERENCES	36

LIST OF FIGURES

FIGURE No	TITLE	PAGE No
1	BLOCK DIAGRAM	8
2	PIC 16F877A	9
3	PIN DIAGRAM	10
4	LCD DISPLAY	12
5	PIN DETAILS	13
6	KEYPAD MATRIX	14
7	GSM MODULE	15
8	POWER SUPPLY	16
9	RS-232	17
10	FLOWCHART	20

1. INTRODUCTION

Real time embedded module has been developed to monitor the functioning of an automatic, the so called smart electronic voting machine. This On the day of election, voters present themselves to the Admin at the main server to get the One Time Password (OTP) that qualifies them to vote. Multiple ballot units are kept at different locations. Admin switches the particular ballot unit on. Then the voter goes to the ballot unit to cast the vote. At the ballot unit, voter is asked for the OTP. When the voter enters the given OTP, it gets verified in the database at the main server. If OTP gets verified correctly, the voter will be authenticated for casting his vote. If OTP is not verified correctly, access will be denied. After authentication process, the list of candidates gets displayed on the LCD. There are two categories of candidate.

2. BLOCK DIAGRAM

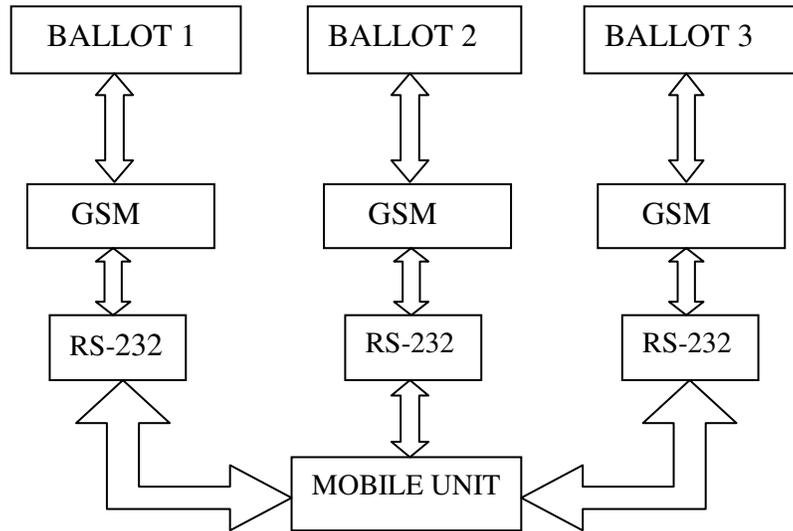


Fig No.2a.Block diagram

3. COMPONENTS

The Major Componets used in Our Project are:

- a) PIC MICROCONTROLLER(16F877A)
- b) LCD DISPLAY
- c) KEYPAD MATRIX
- d) GSM MODULE
- e) POWER SUPPLY
- f) RS-232

3.1. PIC 16F877A

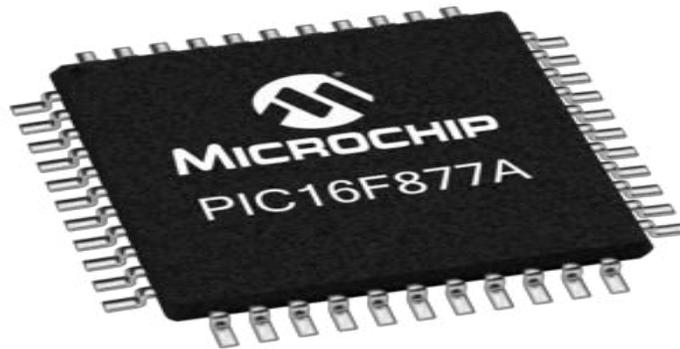


Fig. No.3.1. PIC 16F877A

This powerful (200 nanosecond instruction execution) is easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 40- or 44-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices. The PIC16F877A features 256 bytes of EEPROM data memory, self programming, an ICD, 2 Comparators, 8 channels of 10-bit Analog-to-Digital (A/D) converter, 2 capture/compare/PWM functions, the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I²C™) bus and a Universal Asynchronous Receiver Transmitter (USART). All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications.

3.1a. PIN CONFIGURATION

As it has been mentioned before, there are 40 pins of this microcontroller IC. It consists of two 8 bit and one 16 bit timer. Capture and compare modules, serial ports, parallel ports and five input/output ports are also present in it

PIN DIAGRAM

40-Pin PDIP

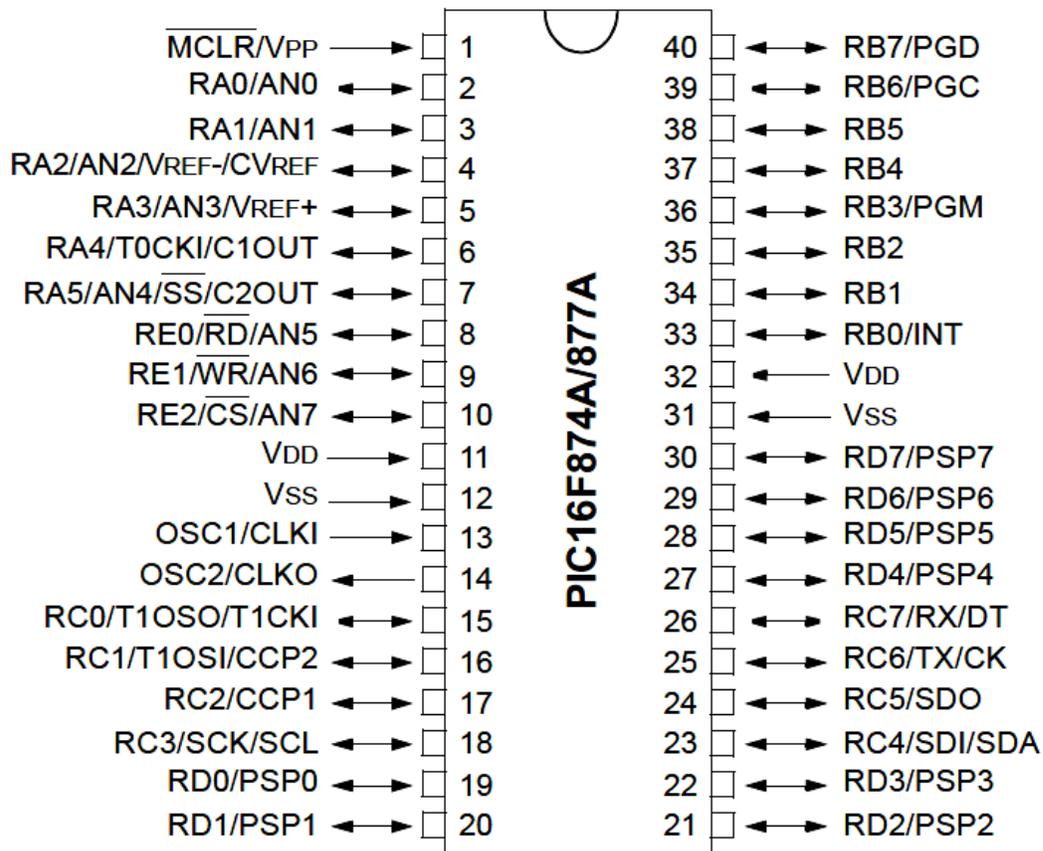


Fig. No.3.1a. Pin diagram of PIC 16F877A

3.1b. FEATURES:

- * High-performance RISC CPU
- * All single cycle instructions except for program branches which are 2 cycle.
- * Operating speed: DC - 20 MHz clock input DC - 200 ns instruction cycle
- * Up to 8K x 14 words of Flash Program Memory,
- * Up to 256 x 8 bytes of EEPROM data memory
- * Pin out compatible to the PIC16C73/74/76/77
- * Interrupt capability -up to 14 internal/external
- * Eight level deep hardware stack
- * Direct, indirect, and relative addressing modes
- * Power-on Reset (POR)
- * Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- * Watchdog Timer (WDT) with its own on-chip RC Oscillator for reliable operation
- * Programmable code-protection
- * Power saving SLEEP mode
- * Selectable oscillator options
- * Low-power, high-speed CMOS EPROM/EEPROM technology
- * Fully static design
- * In-Circuit Serial Programming (ICSP) via two pins
- * Only single 5V source needed for programming capability
- * In-Circuit Debugging via two pins
- * Processor read/write access to program memory
- * Wide operating voltage range: 2.5V to 5.5V
- * High Sink/Source Current: 25 mA
- * Commercial and Industrial temperature ranges

3.2. LCD DISPLAY

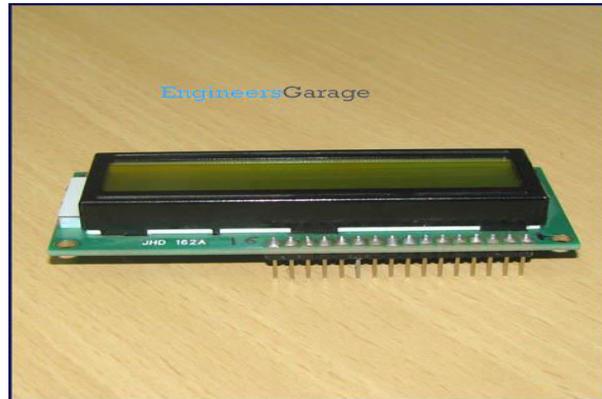


Fig. No.3.2. LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

3.2a. PIN DETAILS OF 2X16 LCD MODULE

PIN No	Name	Function
1	VSS	Ground voltage
2	VCC	+5V
3	VEE	Contrast voltage
4	RS	Register Select 0 = Instruction Register 1 = Data Register
5	R/W	Read/ Write, to choose write or read mode 0 = write mode 1 = read mode
6	E	Enable 0 = start to latch data to LCD character 1 = disable
7	DB0	Data bit 0 (LSB)
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7 (MSB)
15	BPL	Back Plane Light +5V or lower (Optional)
16	GND	Ground voltage (Optional)

Fig. No.3.2a Pin details of LCD module

3.3. KEYPAD MATRIX



Fig. No.3.3. Keypad Matrix

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications. Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a micro-controller. Underneath each key is a push button, with one end connected to one row, and the other end connected to one column. In order for the micro-controller to determine which button is pressed, it first pulls each of the four columns either low or high one at a time, and then polls the states of the four rows. Depending on the states of the rows, the micro controller can tell which button is pressed. 4x4 Keypad pin can directly connect to microcontroller or keypad decoder IC for decoding purpose. The better way to interface a keypad to the PIC is to use a keypad encoder in between the keypad and the microcontroller. One of the Keypad Encoder is the MMC74C922. Simply hook up the entire row

and the column pins directly to the encoder and it will output a HIGH pulse on the Data Available pin whenever a key is pressed.

3.3a. FEATURES:

- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller

3.4. GSM MODULE



Fig. No.3.4. GSM module

This is a plug and play GSM Modem with a simple to interface serial interface. Use it to send SMS, make and receive calls, and do other GSM operations by controlling it through simple AT commands from micro controllers and computers. It uses the highly popular SIM300 module for all its operations.

It comes with a standard RS232 interface which can be used to easily interface the modem to microcontrollers and computers. The modem consists of all the required external circuitry required to start experimenting with the SIM300 module like the power regulation, external antenna, SIM Holder, etc.

3.5. POWER SUPPLY

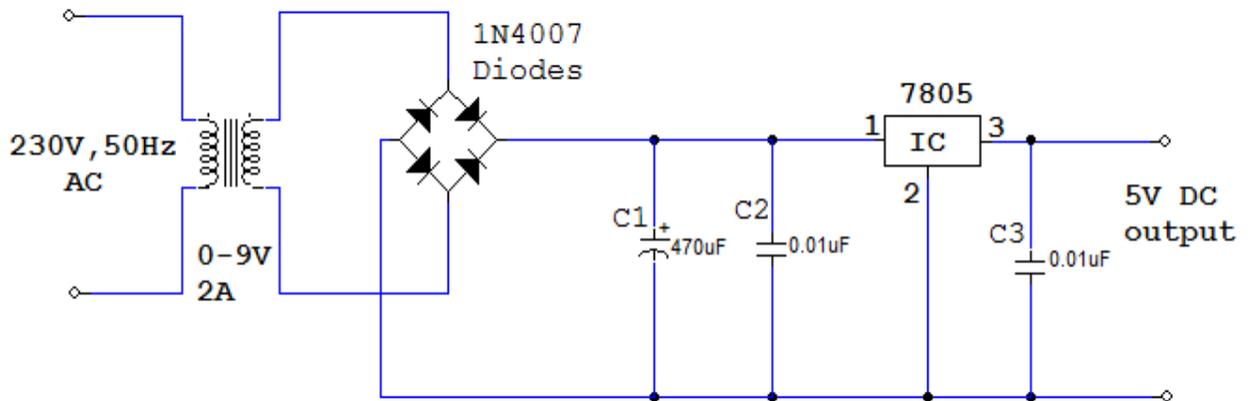


Fig. No.3.5. Power Supply

Five volts power supply w.r.t ground is required for the operation of the microcontroller. The step down transformer steps down the ac supply from the mains and bridge rectifier converts it into dc. This dc is passed through filter circuit to get a more smoothed waveform. Voltage regulator produces a fixed output voltage that remains constant irrespective of changes in its input voltage and load conditions. 7805 voltage regulator is used here, which gives +5 V output which is then given to microcontroller for its operation.

3.6. RS-232

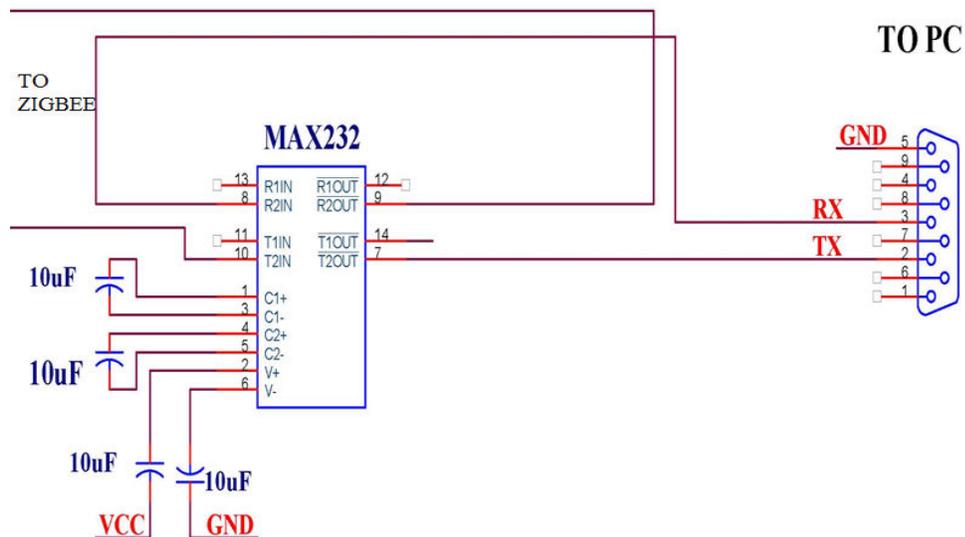


Fig. No.3.6. RS-232

In telecommunications, **RS-232** is a standard for serial communication transmission of data. It formally defines the signals connecting between a DTE (data terminal equipment) such as a computer terminal, and a DCE (data circuit-terminating equipment or data communication equipment), such as a modem. The RS-232 standard is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pinout of connectors. An RS-232 serial port was once a standard feature of a personal computer, used for connections to modems, printers, mice, data storage, uninterruptible power supplies, and other peripheral devices. However, RS-232, when compared to other serial interfaces such as RS-422, RS-485 and Ethernet, is hampered by low transmission speed, short maximum cable length, large voltage swing, large standard connectors, no multipoint capability and limited multidrop capability. In modern personal computers, USB has displaced RS-232 from most of its peripheral interface roles. Many computers no longer come

equipped with RS-232 ports (although some motherboards come equipped with a dedicated COM port jumper that allows the user to install a bracket with a DE-9 port) and must use either an external USB-to-RS-232 converter or an internal expansion card with one or more serial ports to connect to RS-232 peripherals. Nevertheless, RS-232 devices are still used, especially in industrial machines, networking equipment, and scientific instruments.

4. MPLAB IDE VERSION 8.1

MPLAB is a proprietary freeware integrated development environment for the development of embedded applications on PIC and dsPIC microcontrollers, and is developed by Microchip Technology. MPLAB X is the latest edition of MPLAB, and is developed on the NetBeans platform. MPLAB and MPLAB X support project management, code editing, debugging and programming of Microchip 8-bit, 16-bit and 32-bit PIC microcontrollers. MPLAB is designed to work with MPLAB-certified devices such as the MPLAB ICD 3 and MPLAB REAL ICE, for programming and debugging PIC microcontrollers using a personal computer. PICKit programmers are also supported by MPLAB. MPLAB 8.X is the last version of the legacy MPLAB IDE technology, custom built by Microchip Technology in Microsoft Visual C++. MPLAB supports project management, editing, debugging and programming of Microchip 8-bit, 16-bit and 32-bit PIC microcontrollers. MPLAB only works on Microsoft Windows. MPLAB is still available from Microchip's archives, but is not recommended for new projects.

MPLAB supports the following compilers:

- MPLAB MPASM Assembler
- MPLAB ASM30 Assembler

- MPLAB C Compiler for PIC18
- MPLAB C Compiler for PIC24 and dsPIC DSCs
- MPLAB C Compiler for PIC32
- HI-TECH C

5. ALGORITHM

- I Voter verification using aadhar number (as used here)
using the mobile unit.
- II OTP generation from the admin after the entering of aadhar.
- III Reception of OTP from the admin to the voter for the corresponding user name.
- IV OTP authentication back to the server.
- V OTP verification and candidate list display.
- VI Vote casting and result display.

6. FLOWCHART

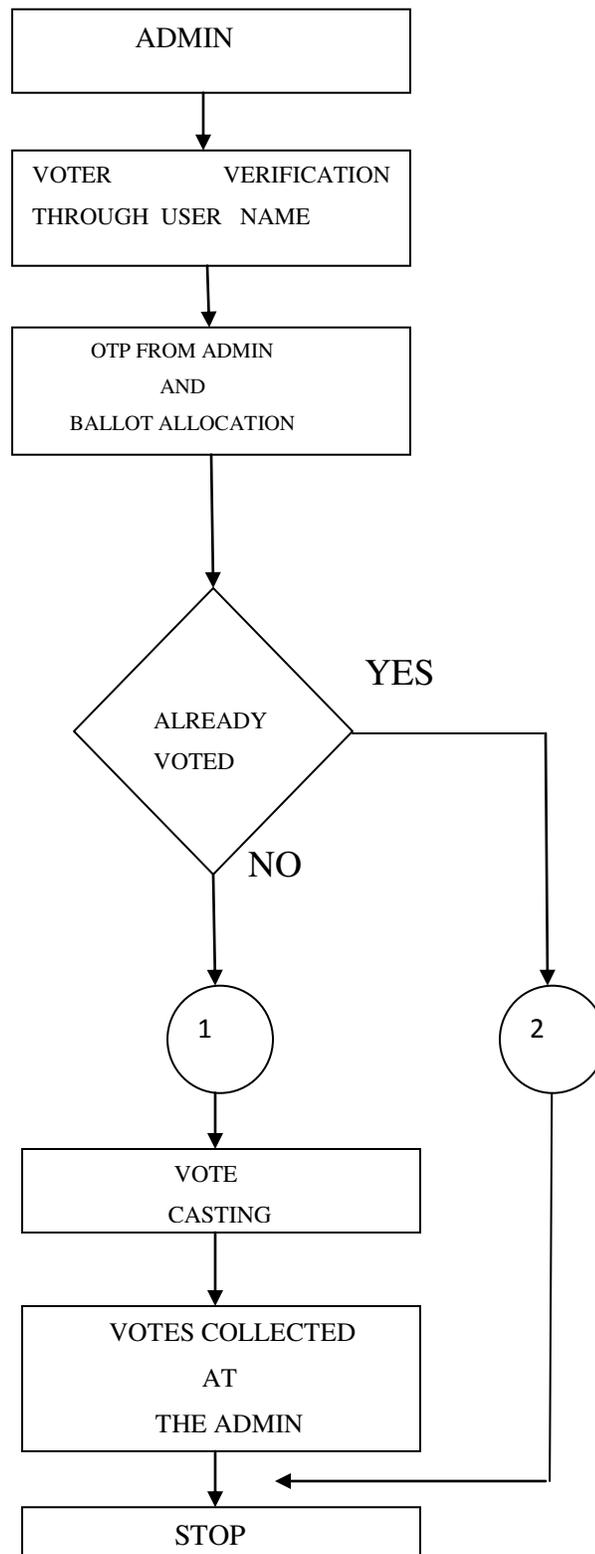


Fig.No.6a. Flowchart

5. WORKING

The functioning of this project is mainly based on the controller which forms the heart of the system. The controller used here is PIC16F877A. The other blocks of the project are interfaced with the controller. 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 numbers and some of the alphabets. The measured parameters will be displayed through the LCD to the driver. As and when the user (voter) wants to vote, he can enter the aadhar number which is being used as the username for authentication purposes. This authentication is carried out by means of OTP generation by the admin at the main server. Once the OTP gets generated it will be sent immediately to the voter via the GSM module used. The communication is done between RS-232 (serial port communication) and the controller. This serves as an intermediate between the voter and the admin. Once the voter receives the OTP he is supposed to enter from his mobile unit and this will be received back at the admin end. The admin checks for the correctness of the OTP from the user. If it matches with the sent OTP then the voter is left open to cast the vote. If not resending of OTP takes place and then the voter will be displayed with a list of candidates contesting in the election. This will get displayed in the LCD module. The voter can cast his vote with the help of keypad matrix. Once he chooses either the candidate or the party, it will be sent to the server and the votes get collected there. This could be further extended for counting of votes and also declaration of results without human intervention.

7. RESULT AND CONCLUSION

Thus a cost effective system is designed successfully for voting and the need for human handled systems is completely ignored here. And also the efficiency in error detection is also higher when compared to the efficiency of the existing electronic voting machines. Time consumption is very less which in turn greatly helps in faster error - free results. As the proposed model is a highly secured one the details of the voters are highly confidential without proper authorization until a need arises.

6. FUTURE SCOPE

The design of this GSM based electronic voting machine can also be extended to voter tracking and could be improved in terms of power consumption using advanced VLSI applications.

8. PROGRAM

```
#include "pic16f877a.h"
#define Keypad_PORT          PORTD
#define Keypad_PORT_Dir    TRISD
unsigned char keypad_read(void);
unsigned char keypad_wait(void);
unsigned char bArrCDN[16] = {0};
unsigned char bKeyCount,bEntryValue = 0;
unsigned char key_get,bStep_0,bStep_1,bStep_2,bStep_3,bStep_4 = 0;
unsigned char bStep_5,bStep_6,bPartySelect = 0;
unsigned char bOTP_Buf[5] = {0x00};
void InitKeypad(void)
```

```

{
    Keypad_PORT    = 0x00;
    Keypad_PORT_Dir = 0x0F;
}
void main()
{
    unsigned char i = 0;

    TRISA =0x3F;
    ADCON0 = 0x00;
    ADCON1 = 0b00000110;
    TRISE =0x00 ;
    PORTB = 0x00;
    TRISB = 0;

    TRISC0 = 1;
    TRISC1 = 1;
    TRISC2 = 1;
    TRISC3 = 0;
    TRISC4 = 0;
    TRISC5 = 0;
    TRISD = 0x0F;
    InitKeypad();
    uartInit();
    LCD_init();
    OPTION_REG = 0b00010110;
    GIE =1;
}

```

```

PEIE = 1;
lcd_disp(1,0,16,"Pls Wait For GSM");
__delay_ms(3000);
__delay_ms(3000);
fnGSMInit();
__delay_ms(2000);
__delay_ms(1000);
lcd_disp(1,0,16,"VOTING MACHINE ");
bEntryValue = 1;

while(1)
{
    if(bEntryValue == 1)
    {
        bEntryValue = 0;
        LCD_command(0x01);
        __delay_ms(500);
        lcd_disp(1,0,16," ENTER AADHAAR ");
        bKeyCount = 0;
        bStep_0 = 1;
        LCD_command(0xC0);
        __delay_ms(500);
        LCD_command(0x14);
        __delay_ms(1000);
        while(bStep_0 == 1)
        {
            key_get = keypad_wait();
            if(key_get == 'D')

```

```

{
    bStep_0 = 0;
}
else if(key_get == 'C')
{
LCD_command(0x01);
__delay_ms(500);
for(i=0;i<16;i++)
bArrCDN[i] = 0;
lcd_disp(1,0,16," ENTER AADHAAR ");
bKeyCount = 0;
bStep_0 == 1;
LCD_command(0xC0);
__delay_ms(500);
LCD_command(0x14);
}
else
{
    bArrCDN[bKeyCount] = key_get;
    Sendchar(bArrCDN[bKeyCount]);
    bKeyCount++;
}
if(bKeyCount >= 12)
{
lcd_disp(1,0,16,"D - OK /C -CLEAR");
}
else

```

```

        {
            __delay_ms(500);
        }
    };
    bKeyCount = 0;
    bStep_1 = 1;
    bStep_0 = 0;
}
else
{
    __delay_ms(10);
}
if(bStep_1 == 1)
{
    bStep_1 = 0;
    bStep_2 = 1;
    lcd_disp(1,0,16,"SMS SENDING.....");
    fnGSMCommand();
    fnU2SendPacket("AADHAAR NUMBER:",15);
    fnU2SendPacket(bArrCDN,12);
    fnSMSSendNow();
    lcd_disp(1,0,16,"SMS OVER.....");
    __delay_ms(1000);
}
else
{
    __delay_ms(10);
}

```

```

}
fnSmsReceiveProcess();
if(g_sbitFlags.bRcvContent == 1)
{
    g_sbitFlags.bRcvContent = 0;
}
else
{
    __delay_ms(10);
}
__delay_ms(100);
if(bRcvFlag == 1)
{
    for(i=0;i<5;i++)
        bOTP_Buf[i] = arrRCVMsg[i+1];

    lcd_disp_full(1,0,5,bOTP_Buf);
    bRcvFlag = 0;
    bStep_3 = 1;
    bStep_2 = 0;
}
else
{
    __delay_ms(10);
}
if(bStep_2 == 1)
{

```

```

        lcd_disp(1,0,16,"WAITING FOR OTP ");
        __delay_ms(2000);
    }
else
{
    __delay_ms(10);
}
if(bStep_3 == 1)
{
    LCD_command(0x01);
    __delay_ms(500);
    lcd_disp_full(1,0,5,bOTP_Buf);
    bKeyCount = 0;
    for(i=0;i<16;i++)
        bArrCDN[i] = 0;
    LCD_command(0xC0);//Second Row cursor
    __delay_ms(500);
    LCD_command(0x14);
    __delay_ms(1000);
    while(bStep_3 == 1)
    {
        key_get = keypad_wait();
        if(key_get == 'D')
        {
            bStep_3 = 0;
            bStep_4 = 1;
        }
    }
}

```

```

else if(key_get == 'C')
{
    LCD_command(0x01);
    __delay_ms(500);
    for(i=0;i<16;i++)
        bArrCDN[i] = 0;
    lcd_disp_full(1,0,5,bOTP_Buf);
    bKeyCount = 0;
    bStep_3 == 1;
    LCD_command(0xC0);
    __delay_ms(500);
    LCD_command(0x14);
}
else
{
    bArrCDN[bKeyCount] = key_get;
    Sendchar(bArrCDN[bKeyCount]);
    bKeyCount++;
}
if(bKeyCount >= 5)
{
    lcd_disp(1,0,16,"D - OK /C -CLEAR");
}
else
{
    __delay_ms(500);
}

```

```

        };
    }
else
{
    __delay_ms(10);
}
if(bStep_4 == 1)
{
    if(strncmp(bArrCDN,bOTP_Buf,5) == 0)
    {
        bStep_5 = 1;
        bStep_3 = 0;
        bStep_4 = 0;
        lcd_disp(1,0,16,"Data Valid    ");
        __delay_ms(2000);
    }
else
{
        bStep_3 = 1;
        bStep_4 = 0;
        lcd_disp(1,0,16,"IN  Valid    ");
        __delay_ms(2000);
    }
}
else
{
    __delay_ms(10);
}

```

```

}

if(bStep_5 == 1)
{
    LCD_command(0x01);
    lcd_disp(1,0,16,"1.BPJ 2.AMK 3.DK ");
    bKeyCount = 0;
    for(i=0;i<16;i++)
        bArrCDN[i] = 0;
    LCD_command(0xC0);//Second Row cursor
    __delay_ms(500);
    LCD_command(0x14);
    __delay_ms(1000);
    while(bStep_5 == 1)
    {
        key_get = keypad_wait();
        if(key_get == 'D')
        {
            bStep_6 = 1;
            bStep_5 = 0;
        }
        else if(key_get == 'C')
        {
            LCD_command(0x01);
            __delay_ms(500);
            for(i=0;i<16;i++)
                bArrCDN[i] = 0;
        }
    }
}

```

```

");

    lcd_disp(1,0,16,"1.BPJ 2.AMK 3.DK

    bKeyCount = 0;
    bStep_5 = 1;
    LCD_command(0xC0);
    __delay_ms(500);
    LCD_command(0x14);
}
else
{
    //bArrCDN[bKeyCount]
    bPartySelect = key_get;
    Sendchar(bPartySelect);
    bKeyCount++;
    __delay_ms(500);
}
if(bKeyCount >= 1)
{
    lcd_disp(1,0,16,"D - OK /C -CLEAR

")

}
else
{
    __delay_ms(500);
}
};
}

```

```

else
{
    __delay_ms(10);
}
if(bStep_6 == 1)
{
    bStep_6 = 0;
    lcd_disp(1,0,16,"SMS SENDING.....");
    fnGSMCommand();
    if(bPartySelect == '1')
    {
        fnU2SendPacket("BPJ Party",9);
    }
    else if(bPartySelect == '2')
    {
        fnU2SendPacket("AMK Party",9);
    }
    else if(bPartySelect == '3')
    {
        fnU2SendPacket(" DK Party",9);
    }
    else
    {
        fnU2SendPacket("Invalid Vote",12);
    }
    fnSMSSendNow();
    lcd_disp(1,0,16,"SMS OVER.....");
}

```

```

        bEntryValue = 1;
        bPartySelect = 0;
    }
    else
    {
        __delay_ms(10);
    }

}

}

void interrupt(void)
{
    if(RCIF == 1)
    {
        //RCIF = 0;
        CREN = 0;
        bRxData = RCREG;
        if(bRxData == '+' || g_sbitStatus.bStartBit == 1)
        {
            if(bRxData != 0x0A)
            {
                if(bCount < 21)
                {
                    arrRcvCmd[bCount] = bRxData;
                }
            }
        }
    }
}

```

```

        bCount++;
        g_sbitStatus.bStartBit = 1;
    }
    else
    {
        g_sbitStatus.bStartBit = 0;
        bCount = 0;
        g_sbitStatus.bProcessbit = 1;
    }
}
}
if(bRxData == '$' || g_sbitStatus.bMsgContent ==1)
{
    if(bRxData != 0x0D)
    {
        arrRCVMsg[bMsgCount] = bRxData;
        bMsgCount++;
        g_sbitStatus.bMsgContent = 1;
    }
    else
    {
        g_sbitStatus.bMsgContent = 0;
        bMsgCount = 0;
        g_sbitStatus.bMsgRcvdbit = 1;
    }
}
}
CREN = 1;

```

```

    }
}
unsigned char keypad_read(void)
{
    KP_R1 = 0;
    KP_R2 = 1;
    KP_R3 = 1;
    KP_R4 = 1;
    __delay_us(30);
    if (KP_C1 == 0) return '0';
    if (KP_C2 == 0) return '4';
    if (KP_C3 == 0) return '8';
    if (KP_C4 == 0) return 'C';
    KP_R1 = 1;
    KP_R2 = 0;
    KP_R3 = 1;
    KP_R4 = 1;
    __delay_us(30);
    if (KP_C1 == 0) return '1';
    if (KP_C2 == 0) return '5';
    if (KP_C3 == 0) return '9';
    if (KP_C4 == 0) return 'D';
    KP_R1 = 1;
    KP_R2 = 1;
    KP_R3 = 0;
    KP_R4 = 1;
    __delay_us(30);

```

```

        if (KP_C1 == 0) return '2';
        if (KP_C2 == 0) return '6';
        if (KP_C3 == 0) return 'A';
        if (KP_C4 == 0) return 'E';
        KP_R1 = 1;
        KP_R2 = 1;
        KP_R3 = 1;
        KP_R4 = 0;
        __delay_us(30);
        if (KP_C1 == 0) return '3';
        if (KP_C2 == 0) return '7';
        if (KP_C3 == 0) return 'B';
        if (KP_C4 == 0) return 'F';
        return 0xFF;
    }
unsigned char keypad_wait(void)
{
    unsigned char c_pressed_key = 0xFF;
    do {
        c_pressed_key = keypad_read();
    }
    while (c_pressed_key == 0xFF);
    while (keypad_read() != 0xFF);
    return c_pressed_key;
}

```

9. REFERENCES

- [1] Alaguvel.R and Gnanavel.G, “Offline and Online E-Voting System with Embedded Security for Real Time Application”,Department of Electrical and Electronics Engineering, V.R.S College of Engineering and Technology, Arasur.
- [2] Ms.Tanzila Afrin and Prof K.J.Satao, “E-Voting System for On Duty Persons Using Public Key Cryptography”,International Journal of Innovative Research in Computer And Communication Engineering, Vol.1, issue 7, September 2013.
- [3] KarthickDatt.A, VishwanathBagalkot, Manjunatha.D and TilakBabu KS, “Biometric Voting Machine”, Department of ECE, MSRIT, Bangalore.
- [4] M. Byrne, K. Greene, and S. Everett, “Usability of Voting Systems: Baseline Data for Paper, Punch Cards, and Lever Machines”, CHI 2007 Proc., Politics & Activism, ACM Press, vol. 1, 1997, pp. 171–180.
- [5] M. Naor and A. Shamir, “Visual Cryptography”, Advances in Cryptology, Eurocrypt 94, Springer, vol. 950, 1995, pp. 1–12.
- [6] J. Benaloh and D. Tuinstra, “Receipt-Free Secret Ballot Elections”, Proc. 26th Ann. ACM Sump. Theory of Computing, 1994, ACM Press, pp. 544–553.
- [7] Gaetti A. Abadan, Khalid A. Dara bah, TawfiqAmari and Omar Qunsul, “Secure National Electronic Voting”, Computer Engineering Department The University of Jordan Amman 11942, Journal of information science and engineering 30, 1339-1364 (2014) 1339.
- [8] Abdalla Al-Ameen and SamaniTalab, “The Technical Feasibility and Security of E-Voting”,Department of Information Technology, University of Neelain, Sudan, The International Arab Journal of Information Technology, Vol. 10, No. 4, July 2013.
- [9] Diponkar Paul and Sobuj Kumar Ray, Member, IACSITInternational, “ A Preview on Microcontroller Based Electronic Voting Machine”,

IACSITInternational Journal of Information and Electronics Engineering, Vol. 3,
No. 2, March 2013.

(R.O.C.) th0040@mail.tut.edu.tw A Secure and Anonymous Electronic Voting
Scheme Based on Key Exchange Protocol.