

# eVote Firmware Code

---

```
/*
JSAL Firmware Code
eVote - Javier Torres, Sylvia Rodriguez, Laura Cruz, Angel Vega
*/

//LCD Setup Definitions*****
#define reset 0x10;    //pin 5.4
#define read 0x08;    //pin 5.3
#define write 0x04;    //pin 5.2
#define chipSel 0x02;  //pin 5.1
#define dataSel 0x01;  //pin 5.0
#define allPins 0xFF   //activate all pins equal to 1
#define displayOff 0x58 //display toggle off, the last bit is 0 negated
                        //turning off the LCD with 1
#define displayOn 0x59 //display toggle on, the last bit is 1 negated turning
                        //on the LCD with 0
#define systemSet 0x40// Initialize device and display
#define scroll 0x44    //Set display start address and display regions
#define csrForm 0x5D   // Set cursor type
#define cgramAdr 0x5C  // Set start address of character generator RAM
#define crsDir1 0x4C   // Set direction of the cursor movement
#define crsDir2 0x4D   // Set direction of the cursor movement
#define crsDir3 0x4E   // Set direction of the cursor movement
#define crsDir4 0x4F   // Set direction of the cursor movement
#define hdotScr 0x5A   // Set horizontal scroll position
#define overlay 0x5B   // Set display overlay format
#define csrW 0x46      // Set cursor address
#define csrR 0x47      // Read cursor address
#define mwrite 0x42    // Write to display memory
#define mread 0x43     // Read from display memory
//LCD Setup Definitions*****
//Functions Declarations*****
void wrCom(void);
void wrData(void);
void iLCD(void);
void clrP4(void);
void clearLCD(void);
void locked(void); //locked will lock screen until computer activates eVote
                  //With UART
void bal1(int); //first ballot for governor
void bal2(int); //second ballot for resident commissioner
void confirm(int, char sel[], int); //confirm previous choice
void sumDisp(int, char gov[], int,
              char com[], int, char par[]); //final
                  //confirmation before printing and UART communications
void party(int);
int enabled(void);
void type(char a[], int);
void lan(void);
void setPos(unsigned char, unsigned char);
```

```

void clrT(void);
void voteSel(int);
int wrTest(char*);
void eVote(void);
void wrDisp(int);
void curOn(void);
void curOff(void);
void wait(void);
void iUart();
void transmit(char);
void connectPC();
void connectPC2();
void uartDelay();
void printS(int);
//Functions Declarations*****

//Global Variables*****
int up = 0x01;
int left = 0x02;
int right = 0x04;
int down = 0x08;
int ok = 0x10;
char a = ' ';
int screen = 0;
//Global Variables*****

//Include Headers*****
#include <msp430x14x.h>
//Include Headers*****

//Main Application*****
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer

    iLCD(); //initialize LCD
    iUart(); //Initialize UART
    setPos(0x00,0xE2);
    type("initializing / inicializando", 28);

    connectPC(); //verify connection to PC
    eVote(); //Main function for eVote
}
//Main Application*****

//Functions*****
void iUart(void)
{
    volatile unsigned int i;
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    P3SEL |= 0x30; // P3.4,5 = USART0 TXD/RXD
    BCSCTL1 |= XTS; // ACLK = LFXT1 = HF XTAL

    do
    {

```

```

IFG1 &= ~OFIFG;           // Clear OSCFault flag
for (i = 0xFF; i > 0; i--); // Time for flag to set
}
while ((IFG1 & OFIFG));    // OSCFault flag still set?

BCSCTL2 |= SELM_3;         // MCLK = LFXT1 (safe)
ME1 |= UTXE0 + URXE0;      // Enable USART0 TXD/RXD
UCTL0 |= CHAR;             // 8-bit character
UTCTL0 |= SSEL0;           // UCLK = ACLK
UBR00=0x00;
UBR10=0x01;
UMCTL0=0x00; /* uart0 2457600Hz 9600bps */
UCTL0 &= ~SWRST;           // Initialize USART statemachine
IE1 |= URXIE0;             // Enable USART0 RX interrupt

_BIS_SR(GIE);              // Enter LPM0 w/ interrupt
}

#pragma vector=USART0RX_VECTOR
__interrupt void usart0_rx (void)
{
    while (!(IFG1 & UTXIFG0)); // USART0 TX buffer ready?
    a = RXBUF0;                // RXBUF0 to TXBUF0
    uartDelay();
}

void transmit(char tr) //transmit a character with the UART
{
    while (!(IFG1 & UTXIFG0));
    TXBUF0 = tr;
    uartDelay();
}

void uartDelay() //create a delay to be used for the Uart functions
{
    int i=0;
    for(i=4000;i>0;i--);
}

void connectPC() //connection test with the PC software
{
    //if unsuccessful, it will remain "initializing"

    while(a!='#')
    {
        transmit('#');
        //the interrupt will change the value of a
    }

    a = '#';
    //connection successful
}

void connectPC2(void) //verify if still connected while eVote is active
{
    transmit('#');
}

```

```

    uartDelay(); //give pc time to respond

    if(a=='#')
    {
        a = ' ';
        if(screen ==11) //recover from error
        {
            screen = 0;
        }
    }
    else
    {

        clrT();
        setPos(0x00,0xE2);
        type("error - favor de ver oficial", 28);

        screen = 11; //screen 11 doesn't exist so the screen will stay stuck
        //until the device is unplugged.

        setPos(0x01,0x22);
        type("error - please see official ", 28);
    }
    //connection successful
}
void iLCD() //Initialize LCD
{
    P4OUT &= 0;    //Sets P4 to low
    P5OUT &= 0;    //Sets P5 to low

    P5DIR |= allPins; //makes P5 all output using define
    P4DIR |= allPins; //all of P4 is output direction

    P5OUT |= reset; //make reset start off high to not have it active
    P5OUT |= read;  //set rd to 1 to disable
    P5OUT |= write; //initialize write to disable

    P4OUT = systemSet; //Send system Set command to initialize LCD
    wrCom();

    P4OUT = 0x30;    //Since the command is set, sending data will
    wrData();        //be understood as data for this specific command
                    //unless a new command is issued. 0x30 is for no invert

    P4OUT = 0x87;    //Character width 1000 XXXX changed from 87
    wrData();        //8 Sets a two frame (16 line inverts after 16 lines
                    //not desired the 7 represents the character
                    //width (7 = 8 pixels wide)

    P4OUT = 0x07;    //Vertical Character height = 0000 XXXX
    wrData();        //same as width, but this value can
                    //get up to 0F unlike the height

    P4OUT |= 0x1F;   //Set C/R, address range for one line of the

```

```

wrData();    //display from 0-239 --- 32 bytes

P4OUT = 0x23;    //Set line range horizontally (screen edges for line) TC/R
wrData();

P4OUT = 0x7F;    //Set Height of the screen in lines L/F 128 lines in height
wrData();

P4OUT = 0x20;    //Virtual Screen address range //Low
wrData();
P4OUT = 0;    //High
wrData();    //The data is received as 0020 with the low
              //first and then high.

P4OUT = scroll; //Send Command for scroll
wrCom();

P4OUT = 0;    //clears the data and begins at
wrData();    //0000h address for layer 1 (text) low = 00

P4OUT = 0;    //High half of the address so it can have 16 bits of info
wrData();

P4OUT = 0x7F;    // 128 lines according to datasheet table
wrData();

P4OUT = 0;    //start layer 2 on address 1000 --- Low
wrData();
P4OUT = 0x10;    //high portion of address
wrData();

P4OUT = 0x7F;    // 128 lines for layer 2 (graphics)
wrData();

P4OUT = hdotScr; //Horizontal Scroll Command
wrCom();
P4OUT = 0x00;    //no change
wrData();

P4OUT = ovlay; //Overlay
wrCom();

P4OUT = 0;
wrData();

clearLCD();    //function to erase everything on screen

P4OUT = csrForm; //Command CSForm to adjust cursor properties
wrCom();

P4OUT = 0x04;    //Set width 0000XXXX 5 pixels
wrData();

P4OUT = 0x86;    //Set Height 1000XXXX 7 pixels
wrData();

```

```

P4OUT = displayOn; //turn on display
wrCom();

P4OUT = 0x14; //turn off layers that are not to be used 0001 0100
wrData(); //0x14 turns on first 2 layers and disactivates the 3rd
//and 4th. Turns cursor on or off. 14 turns cursor off.
//turns on the cursor 16

P4OUT = cgramAdr; //CGRAM, Starts address of Character generator
wrCom();
P4OUT = 0x00; //send address in high and low start at 0400 ---Low
wrData();
P4OUT = 0x04; //high
wrData();
}
//writes data from the P4 port
void wrData()
{
    P5OUT &= ~dataSel; //turn off Ao
    P5OUT &= ~chipSel;
    P5OUT &= ~write;
    P5OUT |= write;
    P5OUT |= chipSel;
    clrP4();
}
void wrCom()//writes the command that is in on the data port P4
{
    P5OUT |= dataSel;
    P5OUT &= ~chipSel;
    P5OUT &= ~write;
    P5OUT |= write;
    P5OUT |= chipSel;
    clrP4();
}

void clrP4() //clears the information in port 9 to zero
{
    P4OUT &= 0x00;
}

void clearLCD()//used to clear the screen in initialization for graphics layer
{
    int i =0;
    int j =0;
    P4OUT |= crsDir1; //Auto Cursor Increment
    wrCom();
    clrT();
    P4OUT = 0; //put zeros into the CGRAM to clear out trash
//28*256 iterations to clear all 7168 spaces in CGRAM
    for(i=896;i>0;i--) //clear CGRAM 3584
    {
        wrData();
        wrData();
        wrData();
        wrData();
    }
}

```

```

    for(j=1024; j>0; j--)//Clear Graphics 4096
    {
        wrData();
        wrData();
        wrData();
        wrData();
    }
}

void clrT() //erase text on screen
{
    int i =0;
    P4OUT |= csrw;    //Drawing Control CSRW
    wrCom();
    P4OUT = 0;        //Address 0000 ---Low
    wrData();
    P4OUT = 0;        //--High
    wrData();
    P4OUT |= mwrite;  //write to display memory
    wrCom();

    for(i=256; i>0 ; i--) //512 spaces for characters 4*128
    {
        P4OUT |=0x20; //ascii space from character map in Datasheet
        wrData();
        P4OUT |=0x20; //ascii space from character map in Datasheet
        wrData();
    }
}

```

```

void eVote() //main eVote device application
{
    /*
    eVote function by: Sylvia Rodriguez Rodriguez
    */

    int lang = 1; //stores the language choice, default spanish (1)
    int i = 0;
    char choiceG[28] = "          ";
    char choiceC[28] = "          ";
    char p[28] = "          ";
    char g[28] = "          ";
    char g1[28]= "anibal acevedo vila    ";
    char g2[28]= "luis fortune          ";
    char g3[28]= "edwin irizarry mora   ";
    char g4[28]= "rogerio figueroa garcia ";
    char g6[28]= "          ";
    char c[28] = "          ";
    char c1[28]= "alfredo salazar       ";
    char c2[28]= "pedro pierluisi       ";
    char c3[28]= "jessica martinez birriel ";
    char c4[28]= "carlos alberto velazquez ";
    char c6[28]= "          ";
    int voto = 0; //stores the voting mode selection: integral or mixed.

```

```

int pos = 1;//stores the position for navigation (up and down)
int yes = 1;
int back = 0;

screen = 0;//stores the actual screen
curOn();//turn on the cursor

while(1)
{
    connectPC2();
    if(screen == 0)//locked screen
    {
        locked();//display lock screen
        while(1)
        {
            if(enabled()==1) // If the kiosk is assigned it will deactivate the lock
            {
                screen = 1;
                break;
            }//end if
        }//end while locked
    }//end of locked screen

    if(screen==1)//language selection screen
    {
        lan();//display language screen
        lang = 1;//default language, spanish
        setPos(0x00,0xE8);
        while(1)
        {
            if (P2IN == left) //left, spanish
            {
                setPos(0x00,0xE8);
                lang = 1;
                wait();
            }//end if left
            else if (P2IN == right)//right, english
            {
                setPos(0x00,0xF8);
                lang = 0;
                wait();
            }//end if right
            else if (P2IN == ok) //ok
            {
                screen = 2;
                wait();
                break;
            }//end if ok
        }//end while language
    }//end of language selection screen

    if(screen ==2)//voting mode selection screen
    {
        voteSel(lang); // display vote mode selection screen, default integral
        setPos(0x00,0xE8);
    }
}

```



```

voto =0;

    while (1)
    {
        if (P2IN == left)//left
        {
            voto =0; //set vote mode to integral
            setPos(0x00,0xE8);
            wait();
        }//end of if left
        else if(P2IN == right)//right
        {
            voto = 1; //set vote to mixed
            setPos(0x00,0xF8);
            wait();
        }//end of if right
        else if (P2IN == ok)//ok
        {
            screen=3;
            wait();
            break;
        }//end of if ok
    }//end of while voting mode selection
} //end of voting mode selection screen

connectPC2();
if(screen==3)//party selection screen
{
    party(lang);    //display party screen
    setPos(0x00,0x4E);
    pos = 1;

    while(1)
    {
        if (P2IN == up && pos>1)//move up if possible
        {
            pos--;
            switch (pos) //move up
            {
                case 1 : //display cursor position 1
                {
                    setPos(0x00,0x4E);
                    break;
                }//end of case position 1
                case 2 ://display cursor position 2
                {
                    setPos(0x00,0x8E);
                    break;
                }//end of case position 2
                case 3 ://display cursor position 3
                {
                    setPos(0x00,0xCE);
                    break;
                }//end of case position 3
                case 4 ://display cursor position 4
                {
                    setPos(0x01,0x0E);

```

```

        break;
    } //end of case position 4
    case 5 : //display cursor position 5
    {
        setPos(0x01,0x4E);
        break;
    } //end of case position 5
    default :
        break;
    } //end switch move up
    wait();
} //end if move up
else if (P2IN == down && pos<6) //move down if possible
{
    pos++;
    switch (pos) //move down
    {
        case 1 : //display cursor position 1
        {
            setPos(0x00,0x4E);
            break;
        } //end of case position 1
        case 2 : //display cursor position 2
        {
            setPos(0x00,0x8E);
            break;
        } //end of case position 2
        case 3 : //display cursor position 3
        {
            setPos(0x00,0xCE);
            break;
        } //end of case position 3
        case 4 : //display cursor position 4
        {
            setPos(0x01,0x0E);
            break;
        } //end of case position 4
        case 5 : //display cursor position 5
        {
            setPos(0x01,0x4E);
            break;
        } //end of case position 5
        case 6 : //display cursor position 6
        {
            setPos(0x01,0xAE);
            break;
        } //end of case position 6

        default :
            break;
    } //end of switch move down
    wait();
} //end if move down

```

```

else if (P2IN == ok) //ok, stores selection into party variable and display confirmation

```

screen

```

{
    wait();
    switch (pos) // store selection
    {
        case 1://party ppd
        {
            p[0] = 'p' ; p[1] = 'p' ; p[2] = 'd' ;
            confirm(lang, p, 3);//display confirm screen
            setPos(0x00,0xE8);

yes=1;
            while (1)
            {
                if(yes ==1)
                {
                    setPos(0x00,0xE8);

                    switch (voto)//voting mode type
                    {
                        case 0: //integral vote,
                        {
                            for(i=0;i<28;i++)
                            {
                                g[i] =

                                }
                                for(i=0;i<28;i++)
                                {
                                    c[i] =

                                }
                                screen = 4;
                                yes = 3;
                                break;
                            }//end case 0 integral vote

                            case 1://mixed vote
                            {
                                yes = 3;
                                screen = 5;
                                break;
                            }

                            default:
                                break;
                        }//end of switch voting mode type
                    }//end if yes = 1

                    else if (yes == 0)
                    {
                        yes=3;
                        screen = 3;
                    }
                }
            }
        }
    }
}

```

store the corresponding governor and commisioner

g1[i];

c1[i];

commisioner

```
setPos(0x00,0xF8);
wait();
//clear governor , party and

p[0] = '' ; p[1] = '' ; p[2] = '' ;
for(i=0;i<28;i++)
{
    g[i] = g6[i];
}
for(i=0;i<28;i++)
{
    c[i] = c6[i];
}

} //end if yes =0
```

party 1

```
if (P2IN == left) //left, confirm
{
    wait();

    setPos(0x00,0xE8);

    yes =1;
} //end of if left, confirm party 1
else if (P2IN == right) //right, undo confirm

{
    wait();
    setPos(0x00,0xF8);
    yes = 0;

} //end if right, undo confirm
else if (P2IN == ok) //ok confirm party 1
{
    wait();
    break;
} //end if ok confirm
} //end while confirm party 1
break;
} //end of case 1

case 2://part pnp
{
    p[0] = 'p' ; p[1] = 'n' ; p[2] = 'p' ;
    confirm(lang, p, 3); //display confirm screen
    setPos(0x00,0xE8);
    yes=1;
    while (1)
    {

        if(yes ==1)
        {
            setPos(0x00,0xE8);
```

store the corresponding governor and commisioner

g2[i];

c2[i];

commisioner

```
switch (voto)//voting mode type
{
    case 0://integral vote,
    {
        for(i=0;i<28;i++)
        {
            g[i] =
        }
        for(i=0;i<28;i++)
        {
            c[i] =
        }
        screen = 4;
        yes = 3;
        break;
    }//end case 0 integral vote

    case 1://mixed vote
    {
        screen = 5;
        break;
    }//end case 1 mixed vote
    default:
    break;
} //end of switch voting mode type
} //end if yes

else if (yes == 0)
{
    yes=3;
    screen = 3;
    setPos(0x00,0xF8);
    wait();
    //clear governor , party and

    p[0] = '' ; p[1] = '' ; p[2] = '' ;
    for(i=0;i<28;i++)
    {
        g[i] = g6[i];
    }
    for(i=0;i<28;i++)
    {
        c[i] = c6[i];
    }

} //end if yes =0

if (P2IN == left)//left, confirm
{
```

```

        wait();
        setPos(0x00,0xE8);

        yes =1;
        }//end of if left, confirm party 3

        if (P2IN == right)//right, undo confirm
        {
            wait();
            setPos(0x00,0xF8);
            yes = 0;
        }//end if right, undo confirm party 2
        else if (P2IN == ok)//ok confirm case 2
        {
            wait();
            break;
        }//end if ok confirm party 2

    }//end while confirm party 2
    break;
} //end of case 2

case 3://party pip
{
    p[0] = 'p' ; p[1] = 'i' ; p[2] = 'p' ;
    confirm(lang, p, 3);//display confirm screen
    setPos(0x00,0xE8);
    yes=1;
    while (1)
    {

        if(yes ==1)
        {
            setPos(0x00,0xE8);

            switch (voto)//voting mode type
            {
                case 0: //integral vote,

                    {
                        for(i=0;i<28;i++)
                        {
                            g[i] =

                        }
                        for(i=0;i<28;i++)
                        {
                            c[i] =

                        }
                        screen = 4;
                        break;
                    } //end case 0 integral vote
            }

```

store the corresponding governor and commisioner

g3[i];

c3[i];

commisioner

```
        case 1://mixed vote
        {
            screen = 5;
            break;
        }
        default:
        break;
    }//end of switch voting mode type
} //end of if yes = 1
else if (yes == 0)
{
    yes=3;
    screen = 3;
    setPos(0x00,0xF8);
```

```
    //clear governor , party and
```

```
    p[0] = ' ' ; p[1] = ' ' ; p[2] = ' ' ;
    for(i=0;i<28;i++)
    {
        g[i] = g6[i];
    }
    for(i=0;i<28;i++)
    {
        c[i] = c6[i];
    }
```

```
    } //end if yes =0
    if (P2IN == left)//left, confirm
    {
        wait();
        setPos(0x00,0xE8);
```

party 3

```
    yes =1;
    } //end of if left, confirm party 3
    else if (P2IN == right)//right, undo confirm
    {
        wait();
        setPos(0x00,0xF8);
        yes = 0;

    } //end if right, undo confirm
    else if (P2IN == ok)//ok confirm party 3
    {
        wait();
        break;
    } //end if ok confirm
```

```
    } //end while confirm party 3
```

```
    break;
} //end of case 3
```

```
case 4://party ppr
```

```

{

p[0] = 'p' ; p[1] = 'p' ; p[2] = 'r' ;
confirm(lang, p, 3); //display confirm screen
setPos(0x00,0xE8);
yes=1;
while (1)
{

    if(yes ==1)
    {
        yes=3;
        setPos(0x00,0xE8);

        switch (voto) //voting mode type
        {
            case 0: //integral vote,

                {
                    for(i=0;i<28;i++)
                    {
                        g[i] =

                    }
                    for(i=0;i<28;i++)
                    {
                        c[i] =

                    }

                    screen = 4;
                    break;
                } //end case 0 integral vote

            case 1: //mixed vote
            {
                screen = 5;
                break;
            }
        } //end of switch voting mode type
    } //end of if yes =1

    else if (yes == 0)
    {
        yes=3;
        screen = 3;
        setPos(0x00,0xF8);
        //clear governor , party and

        p[0] = '' ; p[1] = '' ; p[2] = '' ;

        for(i=0;i<28;i++)
        {

store the corresponding governor and commisioner

g4[i];

c4[i];

commisioner


```



party 4

```
                g[i] = g6[i];
            }
            for(i=0;i<28;i++)
            {
                c[i] = c6[i];
            }

        } //end if yes =0


        if (P2IN == left) //left, confirm
        {
            wait();

            setPos(0x00,0xE8);
            yes =1;
        } //end of if left, confirm party 4
        else if (P2IN == right) //right, undo confirm

        {
            wait();
            setPos(0x00,0xF8);
            yes = 0;

        } //end if right, undo confirm
        else if (P2IN == ok) //ok confirm party 4
        {
            wait();
            break;
        } //end if ok confirm
    } //end while confirm party 4
    break;
} //end of case 4

case 5: //none party selected, blank ballot
{

    p[0] = 'n' ; p[1] = '/' ; p[2] = 'a' ;
    confirm(lang, p, 3); //display confirm screen
    setPos(0x00,0xE8);
    yes=1;
    while (1)
    {

        if(yes ==1)
        {
            setPos(0x00,0xE8);

            switch (voto) //voting mode type
            {

                case 0: //integral vote,

store the corresponding governor and commissioner
```

```

{

p[0] = ' ' ; p[1] =

' ' ; p[2] = ' ' ;

for(i=0;i<28;i++)
{
    g[i] =

}
for(i=0;i<28;i++)
{
    c[i] =

}
screen = 4;
break;
} //end case 0 integral vote

case 1://mixed vote
{
    screen = 5;
    break;

}
default:
break;
} //end of switch voting mode type
} //end of if yes =1

else if (yes == 0)
{
    yes=3;
    screen = 3;
setPos(0x00,0xF8);
    wait();
    //clear governor , party and

    p[0] = ' ' ; p[1] = ' ' ; p[2] = ' ' ;
    for(i=0;i<28;i++)
    {
        g[i] = g6[i];
    }
    for(i=0;i<28;i++)
    {
        c[i] = c6[i];
    }

} //end if yes =0

g6[i];

c6[i];

commisioner

```

party 1

```

        if (P2IN == left)//left, confirm
        {
            wait();
            setPos(0x00,0xE8);

            yes =1;
        }//end of if left, confirm party 1
        else if (P2IN == right)//right, undo confirm

        {
            wait();
            setPos(0x00,0xF8);
            yes = 0;

        }//end if right, undo confirm
        else if (P2IN == ok)//ok confirm party 1
        {
            wait();
            break;
        }//end if ok confirm
    }//end while confirm party 1
    break;
} //end of case 5

case 6://go back to voting mode selection
{
    screen = 2;
    break;
} //end of case 6

} //end switch store selection
break;//break after ok operation exit of while
} //end if ok party selection

} //end of while party selection case 3

} //end of party selection screen

connectPC2();
if(screen==4)//summary screen
{

    sumDisp(lang, g, 28, c, 28, p);
    screen = 0; //set confirm to accept
    setPos(0x01,0xA2);
    while (1)
    {
        if (P2IN == left)//left, confirm summary
        {
            wait();
            setPos(0x01,0xA2);
            screen = 0; //set confirm to accept

        } //end of if confirm summary
        else if(P2IN == right)//right, cancel summary, restart

```

```

        {
            wait();
setPos(0x01,0xB3);
            screen = 1; //set confirm to undo
        } // end of cancel summary, restart

    else if (P2IN == ok) //ok summary accepted
    {

        if(screen == 1) //if you select accept on cancel
        {
            wait();
            break;
        }

        transmit('1'); //header
        for(i=0;i<3;i++) //transmit party
        {
            transmit(p[i]);
        }
        transmit('1'); //tail

        transmit('2'); //header
        for(i=0;i<28;i++)
        {
            transmit(g[i]); //transmit governor
        }
        transmit('2'); //tail

        transmit('3'); //header
        for(i=0;i<28;i++) //transmit commissioner
        {
            transmit(c[i]);
        }
        transmit('3'); //tail

        printS(1);
        while(1)
        {
            if(a=='@')
            {
                a=' ';
                screen=0;
                break;
            }
            else if(a=='%')
            {
                printS(2);
                a=' ';
                screen=9;
                break;
            }

            //send lock signal
        }
    }

```

```

//if you don't accept it returns to screen 1
//lock screen
wait();
break;
} //end of ok summary
} //end of while summary screen

} //end of summary screen

if(screen==5) //governor selection screen
{
    ball(lang);
    pos = 1;
    setPos(0x00,0x5C);
    while(1)
    {
        if (P2IN == up && pos>1) //move up if possible
        {
            pos--;
            switch (pos) //move up
            {
                case 1 : //display cursor position 1
                {
                    setPos(0x00,0x5C);
                    break;
                } //end of case position 1
                case 2 : //display cursor position 2
                {
                    setPos(0x00,0x9C);
                    break;
                } //end of case position 2
                case 3 : //display cursor position 3
                {
                    setPos(0x00,0xDC);
                    break;
                } //end of case position 3
                case 4 : //display cursor position 4
                {
                    setPos(0x01,0x1C);
                    break;
                } //end of case position 4
                case 5 : //display cursor position 5
                {
                    setPos(0x01,0x5C);
                    break;
                } //end of case position 5
                default :
                break;
            } //end switch move up
            wait();
        } //end if move up
        else if (P2IN == down && pos<6) //move down if possible
        {
            pos++;

```

```

switch (pos) //move down
{
    case 1 : //display cursor position 1
    {
        setPos(0x00,0x5C);
        break;
    } //end of case position 1
    case 2 : //display cursor position 2
    {
        setPos(0x00,0x9C);
        break;
    } //end of case position 2
    case 3 : //display cursor position 3
    {
        setPos(0x00,0xDC);
        break;
    } //end of case position 3
    case 4 : //display cursor position 4
    {
        setPos(0x01,0x1C);
        break;
    } //end of case position 4
    case 5 : //display cursor position 5
    {
        setPos(0x01,0x5C);
        break;
    } //end of case position 5
    case 6 : //display cursor position 6
    {
        setPos(0x01,0xBC);
        break;
    } //end of case position 6
    default :
        break;
} //end of switch move down
wait();
} //end if move down
else if (P2IN == ok) //ok, end governor navigation and store selection
{
    wait();
    if(pos==1) //party ppd
    {
        for(i=0;i<28;i++)
        {
            g[i] = g1[i];
        }
        confirm(lang, g, 28); //display confirm screen

        setPos(0x00,0xE8);

        screen = 6;
        while(1)
        {

            if (P2IN == left) //left, confirms governor
            {

```

and go to commisioner screen

back

```
setPos(0x0,0xE8);
    screen = 6;
    wait();
    for(i=0;i<28;i++)
    {
        g[i] = g1[i];
    }
} //end if confirm governor party 1
else if(P2IN == right) //right, cancel and go
{
    setPos(0x00,0xF8);

    screen = 5;
    wait();
    for(i=0;i<28;i++)
    {
        g[i] = g6[i];
    }
} //end of undo confirm governor party 1
else if (P2IN == ok) //ok
{
    wait();
    break;
} //end of ok confirm governor party 1

} //end of while confirm governor party 1
break;
} //end of case 1 party ppd

if(pos==2) //party pnp
{
    for(i=0;i<28;i++)
    {
        g[i] = g2[i];
    }
    confirm(lang, g, 28); //display confirm screen
    setPos(0x00,0xE8);
    screen = 6;
    while(1)
    {
```

and go to commisioner screen

```
if (P2IN == left) //left, confirms governor
{
    setPos(0x00,0xE8);
    screen = 6;
    wait();
    for(i=0;i<28;i++)
    {
        g[i] = g2[i];
    }
} //end if confirm governor party 2
else if(P2IN == right) //right, cancel and go
```

back

```

        {
            setPos(0x00,0xF8);

            screen = 5;
            wait();
            for(i=0;i<28;i++)
            {
                g[i] = g6[i];
            }
        }//end of undo confirm governor party 2
        else if (P2IN == ok)//ok
        {
            wait();
            break;
        }//end of ok confirm governor party 2

    }//end of while confirm governor party 2
    break;
} //end of case 2 party pnp

```

```

if(pos==3)//party pip
{
    for(i=0;i<28;i++)
    {
        g[i] = g3[i];
    }
    confirm(lang, g, 28);//display confirm screen
    setPos(0x00,0xE8);
    screen = 6;
    while(1)
    {
        /*****

```

and go to commisioner screen

```

        if (P2IN == left)//left, confirms governor
        {
            setPos(0x00,0xE8);
            screen = 6;
            wait();
            for(i=0;i<28;i++)
            {
                g[i] = g3[i];
            }
        }//end if confirm governor party 3
        else if(P2IN == right)//right, cancel and go

```

back

```

        {
            setPos(0x00,0xF8);

            screen = 5;
            wait();
            for(i=0;i<28;i++)
            {
                g[i] = g6[i];
            }

```



```

        }//end of undo confirm governor party 3
        else if (P2IN == ok)//ok
        {
            wait();
            break;
        }//end of ok confirm governor party 3

    }//end of while confirm governor party 3
    break;
} //end of case 3 party pip

```

```

if(pos==4)//party ppr
{
    for(i=0;i<28;i++)
    {
        g[i] = g4[i];
    }
    confirm(lang, g, 28);//display confirm screen
    setPos(0x00,0xE8);
    screen = 6;
    while(1)
    {

```

and go to commisioner screen

```

        if (P2IN == left)//left, confirms governor
        {

```

```

            setPos(0x00,0xE8);
            screen = 6;
            wait();
            for(i=0;i<28;i++)
            {
                g[i] = g4[i];
            }

```

back

```

        }//end if confirm governor party 4
        else if(P2IN == right)//right, cancel and go

```

```

        {
            setPos(0x00,0xF8);

            screen = 5;
            wait();
            for(i=0;i<28;i++)
            {
                g[i] = g6[i];
            }

```

```

        }//end of undo confirm governor party 4
        else if (P2IN == ok)//ok
        {
            wait();
            break;
        }//end of ok confirm governor party 4

```

```

    }//end of while confirm governor party 4
    break;

```

```

} //end of case 4 party ppr

else if(pos==5) //write in
{

    wrDisp(lang);
    back = wrTest(choiceG);
    // type(choice, 28);
    if(back==1)
    {
        //stay on same screen
        break;
    }
    else
    {
        for(i=0;i<28;i++)
        {
            g[i] = choiceG[i];
        }

        confirm(lang, choiceG, 28); //display confirm screen

        setPos(0x00,0xE8);

        screen = 6;
        while(1)
        {

            if (P2IN == left) //left, confirms governor
            {

                setPos(0x00,0xE8);
                screen = 6;
                wait();
                for(i=0;i<28;i++)
                {
                    g[i] = choiceG[i];
                }

            } //end if confirm governor party 4
            else if(P2IN == right) //right, cancel and go
            {
                setPos(0x00,0xF8);

                screen = 5;
                wait();
                for(i=0;i<28;i++)
                {
                    g[i] = g6[i];
                }
            } //end of undo confirm governor party 4
            else if (P2IN == ok) //ok
            {
                wait();
                break;
            }
        }
    }
}

and go to commisioner screen

back

```

```

        } //end of ok confirm governor party 4

        } //end of while confirm governor party 4
        break;
    } //end of back check else
} //end of case 5 write in

else if(pos==6) //case 6 back to party selection
{
    screen = 3;
    break;
} //end of case 6 back to party selection

    //end switch pos governor selection storage
} //end if ok, end governor navigation and store selection
} //end of while governor selection screen
} //end of governor selection screen case

connectPC2();
if(screen==6) //commissioner selection screen
{

    bal2(lang);
    pos = 1;
    setPos(0x00,0x5B);
    while(1)
    {
        if (P2IN == up && pos>1) //move up if possible
        {
            pos--;
            switch (pos) //move up
            {
                case 1 : //display cursor position 1
                {
                    setPos(0x00,0x5B);
                    break;
                } //end of case position 1
                case 2 : //display cursor position 2
                {
                    setPos(0x00,0x9B);
                    break;
                } //end of case position 2
                case 3 : //display cursor position 3
                {
                    setPos(0x00,0xDB);
                    break;
                } //end of case position 3
                case 4 : //display cursor position 4
                {
                    setPos(0x01,0x1B);
                    break;
                } //end of case position 4
                case 5 : //display cursor position 5
                {
                    setPos(0x01,0x5C);
                    break;
                }
            }
        }
    }
}

```

```

        } //end of case position 5
    default :
        break;
    } //end switch move up
    wait();
} //end if move up
else if (P2IN == down && pos<6) //move down if possible
{
    pos++;
    switch (pos) //move down
    {
        case 1 : //display cursor position 1
        {
            setPos(0x00,0x5B);
            break;
        } //end of case position 1
        case 2 : //display cursor position 2
        {
            setPos(0x00,0x9B);
            break;
        } //end of case position 2
        case 3 : //display cursor position 3
        {
            setPos(0x00,0xDB);
            break;
        } //end of case position 3
        case 4 : //display cursor position 4
        {
            setPos(0x01,0x1B);
            break;
        } //end of case position 4
        case 5 : //display cursor position 5
        {
            setPos(0x01,0x5C);
            break;
        } //end of case position 5
        case 6 : //display cursor position 6
        {
            setPos(0x01,0xBC);
            break;
        } //end of case position 6
        default :
            break;
    } //end of switch move down
    wait();
} //end if move down
else if (P2IN == ok) //ok, end governor navigation and store selection
{
    wait();
    if(pos==1) //party ppd
    {
        for(i=0;i<28;i++)
        {
            c[i] = c1[i];
        }
        screen = 4;
    }
}

```

[back](#)

[ 29 ]

and go to commisioner screen

back

```
{  
  
    if (P2IN == left)//left, confirms governor  
    {  
  
        setPos(0x00,0xE8);  
        screen = 4;  
        wait();  
        for(i=0;i<28;i++)  
        {  
            c[i] = c2[i];  
        }  
        //break;  
    }//end if confirm commisioner party 2  
    else if(P2IN == right)//right, cancel and go  
    {  
        setPos(0x00,0xF8);  
  
        screen = 6;  
        wait();  
        for(i=0;i<28;i++)  
        {  
            c[i] = c6[i];  
        }  
        //break;  
    }//end of undo confirm commisioner party 2  
    else if (P2IN == ok)//ok  
    {  
        wait();  
  
        break;  
    }//end of ok confirm commisioner party 2  
  
}//end of while confirm commisioner party 2  
  
}//end of case 2 party pnp  
  
else if(pos==3)//party pip  
{  
  
    for(i=0;i<28;i++)  
    {  
        c[i] = c3[i];  
    }  
  
    confirm(lang, c, 28);//display confirm screen  
    setPos(0x00,0xE8);  
    screen = 4;  
    while(1)  
    {
```

and go to commisioner screen

back

```
if (P2IN == left)//left, confirms governor
{
    setPos(0x00,0xE8);
    screen = 4;
    wait();
    for(i=0;i<28;i++)
    {
        c[i] = c3[i];
    }
    //break;
} //end if confirm commisioner party 3
else if(P2IN == right)//right, cancel and go
{
    setPos(0x00,0xF8);

    screen = 6;
    wait();
    for(i=0;i<28;i++)
    {
        c[i] = c6[i];
    }
} //end of undo confirm commisioner party 3
else if (P2IN == ok)//ok
{
    wait();

    break;
} //end of ok confirm commisioner party 3
```

```
} //end of while confirm commisioner party 3
```

```
} //end of case 3 party pip
```

```
else if(pos==4)//party ppr
{
    for(i=0;i<28;i++)
    {
        c[i] = c4[i];
    }
    confirm(lang, c, 28);//display confirm screen

    setPos(0x00,0xE8);

    screen = 4;
    while(1)
    {
```

and go to commisioner screen

```
if (P2IN == left)//left, confirms governor
{

    setPos(0x00,0xE8);
    screen = 4;
```

back

4//back to locked

```
        wait();
        for(i=0;i<28;i++)
        {
            c[i] = c4[i];
        }
        //break;
    }//end if confirm governor party 4
    else if(P2IN == right)//right, cancel and go
    {
        setPos(0x00,0xF8);

        screen = 6;
        wait();
        for(i=0;i<28;i++)
        {
            c[i] = c6[i];
        }
        //break;
    }//end of undo confirm commisioner party

    else if (P2IN == ok)//ok
    {
        wait();

        break;
    }//end of ok confirm commisioner party 4

} //end of while confirm commisioner party 4

} //end of case 4 party ppr

else if(pos==5)//write in
{
    wrDisp(lang);
    back = wrTest(choiceC);
    if(back==1)
    {
        //if back is selected
        break;
    }
    else
    {
        // type(choice, 28);
        for(i=0;i<28;i++)
        {
            c[i] = choiceC[i];
        }

        confirm(lang, choiceC, 28);//display confirm screen

        setPos(0x00,0xE8);

        screen = 4;
        while(1)
        {
```



```

and go to summary screen

back

if (P2IN == left)//left, confirms commissioner
{
    setPos(0x00,0xE8);
    screen = 4;
    wait();
    for(i=0;i<28;i++)
    {
        c[i] = choiceC[i];
    }

    }//end if confirm commissioner party 4
    else if(P2IN == right)//right, cancel and go

    {
        setPos(0x00,0xF8);

        screen = 6;
        wait();
        for(i=0;i<28;i++)
        {
            c[i] = g6[i];
        }
        }//end of undo confirm governor party 4
        else if (P2IN == ok)//ok
        {
            wait();
            break;
        }//end of ok confirm governor party 4

    }//end of while confirm governor party 4
    break;
    }//end of back check for write-in
} //end of case 5 write in

else if(pos==6)//case 6 back to governor selection
{
    screen = 5;
    break;
} //end of case 6 back to commissioner selection

//end switch pos commissioner selection storage
break;
} //end if ok, end commissioner navigation and store selection
} //end of while commissioner selection screen

} // end commissioner selection screen case

} //end of switch screen
} //end of main while
} //end of eVote

int enabled() //for locking and unlocking the device

```

```

{
//UART acquire information of unlock from computer for phase 3
//cancel Uart communications if true
while(1)
{
    if(a == '0')
    {
        return 1;
    }
}
}

//accepts coordinates on where to start typing or drawing.
void setPos(unsigned char posHigh,unsigned char posLow)
{
    P4OUT = csrw;
    wrCom();

    //00 is at the beggining, F0 moves the text to the middle and more to the right
    P4OUT =posLow;
    wrData();
    P4OUT =posHigh;
    wrData();
}

```

```

int wrTest(char a[]) //navigation logic for the write-in
{
//By Laura M. Cruz
// 802-03-1797
//Write-in Navigation

```

```

//Declare Alphabet Array, this stores alphabet options
char Aarray[] ="abcdefghijklmnopqrstuvwxyz<>";
unsigned int AlphaI = 0;

```

```

int i = 0;
int back = 0;
//Declare Blank Array, this will store voters write in text
char Barray[28]="";
unsigned int BlankI = 0;

```

```

//Declare default Alphabet position
unsigned char AHigh = 0x01;
unsigned char ALow = 0x42;

```

```

//Declare default Blank position
unsigned char BHigh = 0x00;
unsigned char BLow = 0xC2;

```

```

//Declare "boolean" variables to know current position
unsigned int OnAlpha = 1;
unsigned int OnEnd = 0;
unsigned int OnBack = 0;

```

```

//Set cursor to default position (letter A)
setPos(AHigh, ALow);

//Turn on cursor
curOn();

//"Infinite" loop that continues to check if button is being pressed
while(1)
{
    //If down arrow is pressed and cursor is on the alphabet
    if(P2IN==down && OnAlpha==1)
    {
        ALow = 0x82;
        setPos(AHigh,ALow);
        OnAlpha=0;
        OnBack=1;
        OnEnd=0;
        wait();
    }

    //If up arrow is pressed and cursor is not on the alphabet
    else if(P2IN== up && OnAlpha==0)
    {
        ALow = 0x42;
        setPos(AHigh,ALow);
        OnEnd=0;
        OnBack=0;
        OnAlpha=1;
        wait();
    }

    //if right arrow is pressed
    else if (P2IN==right)
    {
        //If cursor is on the alphabet
        if(OnAlpha==1)
        {
            //If cursor is on the last element of alphabet, return to first
            if (AlphaI==27){
                AlphaI=0;
                ALow=0x42;
                setPos(AHigh,ALow);
            }
            //Continue moving right
            else{
                AlphaI++;
                ALow++;
                setPos(AHigh,ALow);
            }
        }
        //If cursor is not on the alphabet and is on the "Back button"
        else if (OnAlpha==0 && OnBack ==1)
        {
            OnEnd=1;
            OnBack=0;
        }
    }
}

```

```

        OnAlpha=0;

        //set cursor to END position
        AHigh=0x01;
        ALow=0x89;
        setPos(AHigh,ALow);
    }
    wait();
}

//if left arrow is pressed
else if(P2IN==left)
{
    //If cursor is on the alphabet
    if(OnAlpha==1)
    {
        //If cursor is on first element of the alphabet, move to last
        if(AlphaI==0){
            AlphaI= 27;
            ALow = 0x5D;
            setPos(AHigh,ALow);
        }
        //Continue moving left
        else{
            AlphaI--;
            ALow--;
            setPos(AHigh,ALow);
        }
    }

    //If cursor is not on the alphabet and is on the "End button"
    else if(OnAlpha==0 && OnEnd==1)
    {
        OnEnd=0;
        OnBack=1;

        //set cursor to BACK position
        AHigh=0x01;
        ALow=0x82;
        setPos(AHigh, ALow);
    }
    wait();
}

//if OK is pressed and cursor is on the alphabet
else if(P2IN==ok && OnAlpha==1)
{
    //Move cursor up to the blank
    setPos(BHigh,BLow);

    //if "Backspace" has been selected and horizontal position is more than left boundary
    if (Aarray[AlphaI]=='<' && BLow > 0xC2 )
    {
        //move back horizontally
        BLow--;
        setPos(BHigh,BLow);
    }
}

```

```

//Type blank space
type(" ",1);

//Set position back to alphabet
setPos(BHigh,BLow);

//Erase character from Blank Array
BlankI--;
Barray[BlankI]=' ';
wait();
}
//If Space has been selected and horizontal position is less than right boundary
else if (Aarray[AlphaI]== '>' && BLow < 0xDD )
{
    BLow++;
    setPos(BHigh,BLow);
    type(" ",1);
    Barray[BlankI]=' ';
    BlankI++;
    wait();
}
//If a letter has been selected
else if (BlankI<27 && Aarray[AlphaI]!='<')
{
    //Type text in current position and increase current blank horizontal position
    type(&Aarray[AlphaI],1);
    BLow++;

    //Add typed letter to Blank Array and increase index for next letter
    Barray[BlankI]=Aarray[AlphaI];
    BlankI++;
    wait();
}

//Move cursor back down to alphabet
setPos(AHigh,ALow);
wait();
}

//if OK is pressed and cursor is not on the alphabet
else if (P2IN==ok && OnAlpha==0)
{
    //If cursor is on "End Button"
    if (OnEnd==1)
    {
        //Show write-in result to test
        setPos(0x01,0x00);
        back=0;
        wait();
    }
    //Exit infinite loop
    break;
}
else if (OnBack==1)
{
    //GO back to ballot

```

```

        back = 1;
        wait();
        //Exit infinite loop
        break;
    }
}
} //end
//return Barray;

for(i=0;i<28;i++)
{
    if(i < BlankI )
    {
        a[i] = Barray[i];
    }
    else
    {
        a[i] = ' ';
    }
}
return back;
}

void curOn() //turn on cursor
{
    P4OUT = displayOn; //turn on display
    wrCom();
    P4OUT = 0x16; //turn off layers that are not to be used 0001 0100
    wrData(); //0x14 turns on first 2 layers and disactivates the 3rd
               //and 4th. Turns cursor on or off. 14 turns cursor off.
               //turns on the cursor 16
}

void curOff() //turn off cursor
{
    P4OUT = displayOn; //turn on display
    wrCom();
    P4OUT = 0x14; //turn off layers that are not to be used 0001 0100
    wrData(); //0x14 turns on first 2 layers and disactivates the 3rd
               //and 4th. Turns cursor on or off. 14 turns cursor off.
               //turns on the cursor 16
}

void wait() //waits until the keypad is no longer with a button pressed
{
    while(1)
    {
        if(P2IN==0)
        {
            break;
        }
    }
}

```

```

void wrDisp(int lang) //Display the write in screen in spanish or english
{
    clrT();
    if(lang==1){

        //Show screen title in english
        setPos(0x00,0x4C);
        type("write-in",8);

    }

    else{

        //Show screen title in spanish
        setPos(0x00,0x47);
        type("nominacion directa",18);

    }

    //Show Alphabet and Space on screen
    setPos(0x01,0x42);
    type("abcdefghijklmnopqrstuvwxyz<>",28);

    if(lang==0){

        //Show "END" option
        setPos(0x01,0x82);
        type(" back fin",11);

    }

    else{

        //Show "FIN" option
        setPos(0x01,0x82);
        type(" atras fin",11);

    }

}

void printS(int s) //when summary is sent to be print
{
    //eVote receives an error from the software
    //if the printing was unsuccessful

    clrT();

    if(s==1)
    {
        setPos(0x00,0xE2);
        type("---printing / imprimiendo---", 28);
    }
    else if(s==2)
    {
        setPos(0x00,0xEC);
        type("---error---", 11);

    }

}

```

```

void locked() //locked screen showed when eVote is inactive
{
    clrT();
    setPos(0x00,0xE5);
    type("not active - no activo", 22);
}

void lan() //select language screen
{
    clrT();
    setPos(0x00,0x88);
    type("language -",10);
    type(" idioma", 7);
    setPos(0x00,0xC4);
    type("english",7);
    setPos(0x00,0xD5);
    type("espanol",7);
    setPos(0x00,0xE6);
    type("( )",5);
    setPos(0x00,0xF6);
    type("( )",5);
}

void voteSel(int lang) //screen for showing the choice of mixed or integral votes
{
    clrT();
    if(lang == 0)
    {
        //if Spanish
        setPos(0x00,0x67);
        type("?como deseas votar?", 19);
        setPos(0x00,0xC4);
        type("integral",7);
        setPos(0x00,0xD5);
        type("mixto-cand", 10); //adjust if possible to spell candidatura
    }
    else if(lang == 1)
    {
        //if english
        setPos(0x00,0x87);
        type("how will you vote?", 18);
        setPos(0x00,0xC4);
        type("integral",8);
        setPos(0x00,0xD5);
        type("mixed-cand", 10); //adjust if possible to spell candidatura
    }
    //Display parenthesis regardless of language
    //in keyboard place cursor inbetween on E8
    setPos(0x00,0xE6);
    type("( )",5);
    //in between on F8
    setPos(0x00,0xF6);
    type("( )",5);
}

```



```
void confirm(int lang, char sel[], int len) //choice confirmation screen
```

```
{
    clrT();
    if(lang == 0) //spanish
    {
        setPos(0x00,0x48);
        type("?confirmar seleccion?", 21);
        setPos(0x00,0x85);
        type(sel, len);
        setPos(0x00,0xC7);
        type("si",2);
        setPos(0x00,0xD7);
        type("no", 2);
    }
    else if(lang == 1)
    {
        setPos(0x00,0x45);
        type("confirm selection?", 18);
        setPos(0x00,0x85);
        type(sel, len);
        //show yes no options
        setPos(0x00,0xC7);
        type("yes",3);
        setPos(0x00,0xD7);
        type("no", 2);
    }
    //Parenthesis for selection
    setPos(0x00,0xE6);
    type("( )",5);

    setPos(0x00,0xF6);
    type("( )",5);
}
```

```
void sumDisp(int lang, char gov[], int govLen,
             char com[], int comLen, char par[]) //display summary
```

```
{
    clrT();
    if(lang == 0) //spanish
    {
        setPos(0x00,0x28);
        type("resumen de voto", 16);
        setPos(0x00,0x60);
        type("partido - ",10);
        type(par,3);
        setPos(0x00,0xA0);
        type("gobernador -",12);
        setPos(0x00,0xC0);
        type(gov,govLen);
        setPos(0x01,0x00);
        type("comisionado residente -", 23);
        setPos(0x01,0x20);
        type(com,comLen);
        setPos(0x01,0xA4);
        type("aceptar",7);
        setPos(0x01,0xB5);
    }
```

```

        type("cancelar", 8);
    }
    else if(lang == 1)
    {
        setPos(0x00,0x28);
        type("voting summary", 14);
        setPos(0x00,0x60);
        type("party -", 7);
        type(par,3);
        setPos(0x00,0xA0);
        type("governor -",10);
        setPos(0x00,0xC0);
        type(gov,govLen);
        setPos(0x01,0x00);
        type("resident commissioner -", 23);
        setPos(0x01,0x20);
        type(com,comLen);
        setPos(0x01,0xA4);
        type("accept",6);
        setPos(0x01,0xB5);
        type("cancel", 6);
    }
}

```

```

void ball(int lang) //governor's ballot
{
    clrT();
    setPos(0x00,0x00);
    if(lang == 0) //spanish
    {
        type("gobernador :", 12);
        //candidates on 40, 80, C0, 100, 140 - write-in
        setPos(0x01,0x45);
        type("nominacion directa ( )", 26);
        setPos(0x01,0xA5);
        type("atras ( )", 26);
    }
    else
    {
        type("governor :", 10);
        setPos(0x01,0x45);
        type("write-in ( )", 26);
        setPos(0x01,0xA5);
        type("back ( )", 26);
    }
    //candidates
    setPos(0x00,0x40);
    type("ppd - anibal acevedo vila ( )", 31);
    setPos(0x00,0x80);
    type("pnp - luis fortune ( )", 31);
    setPos(0x00,0xC0);
    type("pip - edwin irizarry mora ( )", 31);
    setPos(0x01,0x00);
    type("ppr - rogelio figueroa ( )", 31);
}

```

```

void bal2(int lang) //resident commisioners ballot
{
    clrT();
    setPos(0x00,0x00);
    if(lang == 0)
    {
        //spanish
        type("comisionado residente :", 23);
        setPos(0x01,0x45);
        type("nominacion directa ( )", 26);
        setPos(0x01,0xA5);
        type("atras ( )", 26);
    }
    else
    {
        type("resident commissioner :", 24);
        setPos(0x01,0x45);
        type("write-in ( )", 26);
        setPos(0x01,0xA5);
        type("back ( )", 26);
    }
    //candidates
    setPos(0x00,0x40);
    type("ppd - alfredo salazar ( )", 30);
    setPos(0x00,0x80);
    type("pnp - pedro pierluisi ( )", 30);
    setPos(0x00,0xC0);
    type("pip - jessica martinez ( )", 30);
    setPos(0x01,0x00);
    type("ppr - carlos alberto ( )", 30);
}

void party(int lang)
{
    clrT();
    setPos(0x00,0x00);
    if(lang == 0) //spanish
    {
        type("partido :", 9);
        setPos(0x01,0x45);
        type("ninguno( )", 12);
        setPos(0x01,0xA5);
        type("atras ( )", 12);
    }
    else
    {
        type("party :", 7);
        setPos(0x01,0x45);
        type("none ( )", 12);
        setPos(0x01,0xA5);
        type("back ( )", 12);
    }
    //party
    setPos(0x00,0x45);
    type("ppd ( )", 12);
}

```

```

setPos(0x00,0x85);
type("pnp ( )", 12);
setPos(0x00,0xC5);
type("pip ( )", 12);
setPos(0x01,0x05);
type("ppr ( )", 12);
}

void type(char a[], int size) //type letters onto the screen
{

```

```

    int i=0;
    int len = size;
    P4OUT |= mwrite; //write to display memory
    wrCom();

    for(i = 0; i<len ;i++)
    {
        switch (a[i])
        {
            case 'a' :
                P4OUT = 0x41; //hex code for an A
                wrData();
                break;

            case 'b' :
                P4OUT = 0x42; //hex code for letter B
                wrData();
                break;

            case 'c' :
                //hex code for letter C
                P4OUT = 0x43;
                wrData();
                break;

            case 'd' :
                //hex code for letter D
                P4OUT = 0x44;
                wrData();
                break;

            case 'e' :
                //hex code for letter E
                P4OUT = 0x45;
                wrData();
                break;

            case 'f' :
                //hex code for letter F
                P4OUT = 0x46;
                wrData();
                break;

            case 'g' :
                //hex code for letter G
                P4OUT = 0x47;

```

```

wrData();
break;

case 'h' :
//hex code for letter H
P4OUT = 0x48;
wrData();
break;

case 'i' :
//hex code for letter I
P4OUT = 0x49;
wrData();
break;

case 'j' :
//hex code for letter J
P4OUT = 0x4A;
wrData();
break;

case 'k' :
//hex code for letter K
P4OUT = 0x4B;
wrData();
break;

case 'l' :
//hex code for letter L
P4OUT = 0x4C;
wrData();
break;

case 'm' :
//hex code for letter M
P4OUT = 0x4D;
wrData();
break;

case 'n' :
//hex code for letter N
P4OUT = 0x4E;
wrData();
break;

case 'o' :
//hex code for letter O
P4OUT = 0x4F;
wrData();
break;

case 'p' :
//hex code for letter P
P4OUT = 0x50;
wrData();
break;

```

```

case 'q' :
//hex code for letter Q
P4OUT = 0x51;
wrData();
break;

case 'r' :
//hex code for letter R
P4OUT = 0x52;
wrData();
break;

case 's' :
//hex code for letter S
P4OUT = 0x53;
wrData();
break;

case 't' :
//hex code for letter T
P4OUT = 0x54;
wrData();
break;

case 'u' :
//hex code for letter U
P4OUT = 0x55;
wrData();
break;

case 'v' :
//hex code for letter V
P4OUT = 0x56;
wrData();
break;

case 'w' :
//hex code for letter W
P4OUT = 0x57;
wrData();
break;

case 'x' :
//hex code for letter X
P4OUT = 0x58;
wrData();
break;

case 'y' :
//hex code for letter Y
P4OUT = 0x59;
wrData();
break;

case 'z' :

```

```

//hex code for letter Z
P4OUT = 0x5A;
wrData();
break;

case '*' :
//hex code for letter *
P4OUT = 0x2A;
wrData();
break;

case '(' :
//hex code for letter (
P4OUT = 0x28;
wrData();
break;

case ')' :
//hex code for letter )
P4OUT = 0x29;
wrData();
break;

case '&' :
//hex code for letter ' apostrophe
P4OUT = 0x27;
wrData();
break;

case '-' :
//hex code for letter -
P4OUT = 0x2D;
wrData();
break;

case '?' :
//hex code for a ?
P4OUT = 0x3F;
wrData();
break;

case ' ' :
//hex code for a space
P4OUT = 0x20;
wrData();
break;

case '>' :
//writein space symbol
P4OUT = 0x7E;
wrData();
break;

case '<' :
P4OUT = 0x7F;
wrData();

```

```
        break;

        case ':':
            P4OUT = 0x3A;
            wrData();
            break;

        case '/':
            P4OUT = 0x2F;
            wrData();
            break;

        default :
            break;
    }
}
```