

```

#include <xc.h>

//defining clock frequency for the PIC controller
#define _XTAL_FREQ 6000000

//function declaration
void ADC_LCD_PIC_init(void);
void LCD_command(unsigned char);
void LCD_data(unsigned char);
void LCD_output(unsigned int);

//variable initialization
unsigned char L_value, H_value;
long value, voltage;
unsigned char num_arr[10]; // for storing LCD_output values

```

```

void main(void) {
    ADC_LCD_PIC_init();
    while (1) {
        ADCON0 |= 0x04;
        while (ADCON0 & 0x04);
        L_value = ADRESL;
        H_value = ADRESH;
        value = ((unsigned int) H_value << 8) + (unsigned int) L_value;
        voltage = value * 48 / 1023;
        LCD_command(0X80);
        if (voltage == 0) LCD_data(0x30);
        else LCD_output(voltage);
        LCD_data(0x56);
        LCD_data(0x20);
        __delay_ms(1000);
    }
}

```

```

void ADC_LCD_PIC_init(void) {
    //PIC_LCD initialization
    TRISC = 0x00; // setting port C as output //RS,Enable pin
    TRISD = 0x00; // setting port D as output// data pins
    //PIC_ADC initialization
    TRISA = 0x01; //0000 0001 setting RA0 as input - AN0 channel for ADC
    //bit 0(1) - ADC power ON
    //bit 3 to 5 (000)- selecting channel AN0
    //bit 6,7(10)- for choosing fosc/32
    ADCON0 = 0x81; //1000 0001
    //bit 3to0(1110)- AN0 alone as analog input other channel as digital
    //bit 6(0)- for choosing fosc/32
    //bit 7(1)-1 = Right justified. Six (6) Most Significant bits of ADRESH are read as ?0?.
    ADCON1 = 0x8E; //1000 1110
    //LCD initialization
    LCD_command(0x30);
    __delay_ms(50);
    LCD_command(0x30);
    __delay_ms(50);
    LCD_command(0x30);
    __delay_ms(50);
    LCD_command(0x38); // function set (number of lines and 5*7 matrix)
    while (RD7 == 1); //checking busy flag connected D7 bit for ready condition
    LCD_command(0x08); //display OFF
    while (RD7 == 1); //checking busy flag connected D7 bit for ready condition
    LCD_command(0x01); //clear display
    while (RD7 == 1); //checking busy flag connected D7 bit for ready condition
    LCD_command(0x06); //entry mode
    while (RD7 == 1); //checking busy flag connected D7 bit for ready condition

    LCD_command(0x0E); //display ON,cursor display ON
    while (RD7 == 1);
}

void LCD_output(unsigned int n) {
    unsigned char result, i = 0;
    while (n != 0) {
        result = n - ((n / 10)*10);
        num_arr[i] = result;
        i++;
        n /= 10;
    }

    num_arr[i] = '\0';

    i -= 1;
    for (int j = i; j >= 0; j--) {
        LCD_data(0x30 + num_arr[j]);
    }
}

```

```

□ //RS pin - LOW (for sending command)-connected to RC3)
  //Enable pin to go high to low(connected to RC0)

□ void LCD_command(unsigned char hex) {
  PORTC &= 0xF7; //1111 0111(mask bit for AND)clearing RC3
  PORTD = hex;
  PORTC |= 0x01; //0000 0001(mask bit for OR)setting RC0
  PORTC &= ~0x01; //1111 1110(mask bit for AND)clearing RC0
  __delay_ms(50);
}

□ //RS pin - HIGH (for sending data)-connected to RC3)
  //Enable pin to go high to low(connected to RC0)

□ void LCD_data(unsigned char hex) {
  PORTC |= 0x08; //0000 1000(mask bit for OR)setting RC3
  PORTD = hex;
  PORTC |= 0x01; //0000 0001(mask bit for OR)setting RC0
  PORTC &= ~0x01; //1111 1110(mask bit for AND)clearing RC0
  __delay_ms(50);
}

```