

EXPT. 5_A

```
// Program to flash LEDs
#include <p18f4550.h>

void delay(unsigned int time)
{
    unsigned int i,j;
    for(i=0;i<time;i++)
        for(j=0;j<5000;j++);
}

void main(void)
{
    TRISB = 0x00;
    while(1)
    {
        PORTB=0x0F;
        delay(100);
        PORTB=0xF0;
        delay(100);
    }
}
```

EXPT. 5_B

```
#include<P18F4550.h>
void delay()
{
    unsigned int i;
    for(i=0;i<30000;i++);
}
void main()
{
    unsigned char i, key = 0;
    TRISB = 0x00;           //LED pins as output
    //LATB = 0x00;
    ADCON1 = 0x0F;         //set pins as Digital
    TRISAbits.TRISA2 = 1;   //set RA2 as input
    TRISAbits.TRISA3 = 1;   //set RA3 as input

    TRISAbits.TRISA5 = 0;   //set buzzer pin RA5 as output
    TRISAbits.TRISA4 = 0;   //set relay pin RA4 as output
    while(1)
    {
        //LATABits.LA2 = 1;
        //LATABits.LA3 = 1;

        if(PORTAbits.RA2 == 0) key =0;   //If button1 pressed
        if(PORTAbits.RA3 == 0) key =1;   //If button2 pressed

        if(key == 0)
        {
            PORTAbits.RA4 = 1;           //Relay OFF
            PORTAbits.RA5 = 0;           //Buzzer OFF
            for(i=0;i<8;i++)             //Chase LED right to left
            {
                PORTB = 1<<i;
                delay();
                PORTB = 0x00;
                delay();
            }
        }
        if(key == 1)
        {
            PORTAbits.RA4 = 0;           //Relay ON
            PORTAbits.RA5 = 1;           //Buzzer ON
            for(i=7;i> 0;i--)            //Chase LED left to right
            {
                PORTB = 1<<i;
                delay();
                PORTB = 0x00;
                delay();
            }
        }
    }
}
```

1. LCD INTERFACING WITH PIC

```
#include <p18f4550.h>

#define LCD_EN LATAbits.LA1
#define LCD_RS LATAbits.LA0
#define LCDPORT LATB

void lcd_delay(unsigned int time)
{
    unsigned int i, j ;

    for(i=0;i<time;i++)
    {
        for(j=0;j<100;j++);
    }
}

void SendInstruction(unsigned char command)
{
    LCD_RS =0;// RS low : Instruction
    LCDPORT = command;
    LCD_EN =1;// EN High
    lcd_delay(10);
    LCD_EN =0;// EN Low; command sampled at EN falling edge
    lcd_delay(10);
}

void SendData(unsigned char lcddata)
{
    LCD_RS =1;// RS HIGH : DATA
    LCDPORT =lcddata;
    LCD_EN =1;// EN High
    lcd_delay(10);
    LCD_EN =0;// EN Low; data sampled at EN falling edge
    lcd_delay(10);
}

unsigned char*String1 ="PIC18F4550";
unsigned char*String2 ="ENTC dept";

void main(void)
{
    ADCON1 =0x0F;
    TRISB =0x00;          //set data port as output
    TRISAbits.RA0=0;      //RS pin
    TRISAbits.RA1=0;      // EN pin
    // TRISAbits.RA2 = 0;
    SendInstruction(0x38);    //8 bit mode, 2 line,5x7 dots
    SendInstruction(0x06);    // entry mode
    SendInstruction(0x0C);    //Display ON cursor OFF
    SendInstruction(0x01);    //Clear display

    SendInstruction(0x80);    //set address to 1st line
    while(*String1)
    {
        SendData(*String1);
        String1++;
    }
    SendInstruction(0xC0);    //set address to 2nd line
    while(*String2)
    {
        SendData(*String2);
        String2++;
    }
    while(1);
}
```

2. LCD INTERFACING WITH PIC

```
#include<p18f4550.h>

#define LCD_EN PORTCbits.RC1
#define LCD_RS PORTCbits.RC0

void delay()

{
    unsigned int i;
    for (i=0;i<5000;i++);
}

unsigned char string1[]={'E','&','T','C'};

void Sendcommand(unsigned char command)
{
    LCD_RS=0;
    delay();
    LCD_EN=1;
    delay();
    PORTB= command;
    delay();
    LCD_EN=0;
    delay();

}

void Senddata(unsigned char data)
{
    LCD_RS=1;
    delay();
    LCD_EN=1;
    delay();
    PORTB= data;
    delay();
    LCD_EN=0;
    delay();
}
```

```
void main()
{
    unsigned char i,j;
    TRISB=0x00;
    TRISCbits.RC1=0;
    TRISCbits.RC0=0;
    // Initialize LCD
    Sendcommand(0x38);
    Sendcommand(0x0E);
    Sendcommand(0x01);
    while(1)
    {
        Sendcommand(0x80);
        for(i=0;i<4;i++)
        {
            j=string1[i];
            Senddata(j);
        }
    }
}
```

ADC INTERFACING WITH PIC

```
//Adc interfacing with PIC

#include <p18f4550.h>
#include<stdio.h>
#define LCD_EN LATAbits.LA1
#define LCD_RS LATAbits.LA0
#define LCDPORT LATB

void lcd_delay(unsigned int time)
{
    unsigned int i, j ;

    for(i=0;i<time;i++)
    {
        for(j=0;j<50;j++);
    }
}

void SendInstruction(unsigned char command)
{
    LCD_RS =0;          // RS low : Instruction
    LCDPORT = command;
    LCD_EN =1;          // EN High
    lcd_delay(10);
    LCD_EN =0;          // EN Low; command sampled at EN falling edge
    lcd_delay(10);
}

void SendData(unsigned char lcddata)
{
    LCD_RS =1;          // RS HIGH : DATA
    LCDPORT =lcddata;
    LCD_EN =1;          // EN High
    lcd_delay(10);
    LCD_EN =0;          // EN Low; data sampled at EN falling edge
    lcd_delay(10);
}

void InitLCD(void)
{
    ADCON1 =0x0F;
    TRISB =0x00;//set data port as output
    TRISAbits.RA0=0;//RS pin
    TRISAbits.RA1=0;// EN pin

    SendInstruction(0x38);    //8 bit mode, 2 line,5x7 dots
    SendInstruction(0x06);    // entry mode
    SendInstruction(0x0C);    //Display ON cursor OFF
    SendInstruction(0x01);    //Clear display
    SendInstruction(0x80);    //set address to 0
}

void ADCInit(void)
{
    TRISEbits.RE1=1;          //ADC channel 6 input
    TRISEbits.RE2=1;          //ADC channel 7 input

    ADCON1 =0b00000111;      //Ref voltages Vdd&Vss; AN0 -
    AN7 channels Analog
    ADCON2 =0b10101110;      //Right justified; Acquisition
    time 12T; Conversion clock Fosc/64
}
```

```

unsigned short Read_ADC(unsigned char Ch)
{
    ADCON0 = 0b00000001|(Ch<<2);    //ADC on; Select channel;
    GODONE = 1;                        //Start Conversion

    while(GO_DONE == 1);    //Wait till A/D conversion is complete
    return ADRES;           //Return ADC result
}

void DisplayResult(unsigned short ADCVal)
{
    unsigned char i, text[16];
    unsigned short tempv;
    tempv = ADCVal;

    SendInstruction(0x80);    //set to 1st line
    for(i=0; i<10; i++)    //Display the 10 bit ADC result
        on LCD
        {
            if(tempv & 0x200)
            {
                SendData('1');
            }
            else
            {
                SendData('0');
            }
            tempv = tempv << 1;
        }

    ADCVal = (5500/1024)*ADCVal;    //Convert binary data to
    mV;    1 bit <=> (5500/1024)mV
    sprintf(text, "ADC value=%4dmv", ADCVal);    //Convert integer data
    to string

    SendInstruction(0xC0);    //set to 2nd line
    for(i=0; i<16; i++)    //Display string on LCD
    {
        SendData(text[i]);
    }
}

void main()
{
    unsigned short Ch_result;

    TRISB = 0x00;    //PORTB connected to LCD is
    output
    ADCInit();
    InitLCD();
    while(1)
    {
        Ch_result = Read_ADC(7);
        DisplayResult(Ch_result);
        lcd_delay(1000);
    }
}

```

GENERATION OF PWM FOR DC MOTOR CONTROL

```
// Generation of PWM signal for DC motor control.(PIC)
#include<pl8f4550.h>
unsigned char count=0;
bit TIMER,SPEED_UP;

void timer2Init(void)
{
    T2CON    =    0b00000010;           //Prescalar = 16; Timer2 OFF
    PR2      =    0x95;                 //Period Register
}

void delay(unsigned int time)
{
    unsigned int i,j;
    for(i=0;i<time;i++)
        for(j=0;j<1000;j++);
}

void main(void)
{
    unsigned int i;
    TRISCbits.TRISC1    =0;             //RC1 pin as output
    TRISCbits.TRISC2    =0;             //CCP1 pin as output
    LATCbits.LATC1      =0;
    CCP1CON    =    0b00111100;         //Select PWM mode; Duty cycle
LSB CCP1CON<4:5> = <1:1>
    CCPR1L     =    0x0F;               //Duty cycle 10%
    timer2Init();                       //Initialise Timer2
    //Interrupt_Init();                 //Initialise interrupts

    //SPEED_UP = 1;
    TMR2ON =1;                          //Timer2 ON

    while(1)                            //Loop forever
    {
        for(i=15;i<150;i++)
        {
            CCPR1L =i;
            delay(100);
        }
        for(i=150;i>15;i--)
        {
            CCPR1L =i;
            delay(100);
        }
    }
}
```


LCD INTERFACING WITH 8051

```
#include<reg51.h>
sbit rs=P2^2;
sbit rw= P2^1;
sbit en= P2^0;
void DelayMs(delay)
{
    int i,j;
    for(i=0;i<delay;i++)
    {
        for(j=0;j<100;j++);
    }
}

void write_lcd_data(value)
{
    P1=value;
    DelayMs(250);
    rs=1;
    rw=0;
    en=1;
    DelayMs(10);
    en=0;
}

void write_lcd_command(value)
{
    P1=value;
    DelayMs(250);
    rs=0;
    rw=0;
    en=1;
    DelayMs(10);
    en=0;
}

void main(void)
{
    P1=0x00;
    P2=0x00;
    write_lcd_command(0x38);           // function set
    write_lcd_command(0x08);           // display off
    write_lcd_command(0x01);           //display clear
    write_lcd_command(0x06);           //entry mode set
    //write_lcd_command(0x08);
    //write_lcd_command(0x0C);
    write_lcd_command(0x0F);           // display on

    write_lcd_command(0x81);           // set address counter value
    write_lcd_data('8');
    write_lcd_data('0');
    write_lcd_data('5');
    write_lcd_data('1');
    write_lcd_data('+');
    write_lcd_data('L');
    write_lcd_data('C');
    write_lcd_data('D');

    write_lcd_command(0xc0);           //set address counter value
    write_lcd_data('E');
    write_lcd_data('&');
    write_lcd_data('T');
    write_lcd_data('C');
    write_lcd_data('d');
    write_lcd_data('e');
    write_lcd_data('p');
    write_lcd_data('t');
```

```
        // DelayMs(250);  
        while(1)  
        {  
  
        }  
}
```