

1st expt : Simple programs on Memory data transfer

Internal to internal memory data transfer

```
ORG 00h
MOV R0,#30h
MOV R1,#50h
MOV R2,#0Ah
BACK:
MOV A,@R0
MOV @R1,A
INC R0
INC R1
DJNZ R2,BACK
END
```

Internal to external memory data transfer

```
ORG 00h
MOV R0,#30h
MOV R1,#0Ah
MOV DPTR,#0100h
BACK:
MOV A,@R0
MOVX @DPTR,A
INC R0
INC DPTR
DJNZ R1,BACK
END
```

External to internal memory data transfer

```
ORG 00h
MOV R0,#30h
MOV R1,#0Ah
MOV DPTR,#0100h
BACK:
MOVX A,@DPTR
MOV @R0,A
INC R0
INC DPTR
DJNZ R1,BACK
END
```

External to external memory data transfer

```
org 00h
mov DPTR,#0100h
mov R0, #00h
mov R1,#05h
mov R2,#0Ah
AGAIN:movx A,@DPTR
push DPL
push DPH
mov DPL,R0
mov DPH,R1
movx @DPTR,A
pop DPH
pop DPL
inc DPTR
inc R0
DJNZ R2,AGAIN
End
```

2nd expt: LED interfacing with 8051
Microcontroller.

1)LED Flashing

```
#include<AT89X51.h>
```

```
void Delay(void);
```

```
void main (void)
```

```
{
```

```
while(1)
```

```
{
```

```
P2 = 0x55;
```

```
Delay();
```

```
P2 = 0xAA;
```

```
Delay();
```

```
} }
```

```
void Delay(void)
```

```
{ int j;
```

```
int i;
```

```
for(i=0;i<10;i++)
```

```
{
```

```
for(j=0;j<10000;j++)
```

```
{ } } }
```

2) BCD Counter

```
#include<reg51.h>
```

```
void Delay(void);
```

```
void main (void)
```

```
{
```

```
unsigned char count=0x00, x,
```

```
d1,d2,d3;
```

```
while(1)
```

```
{
```

```
x=count/10;
```

```
d1= count%10;
```

```
d2= x%10;
```

```
d3= x/10;
```

```
P2 = d1;
```

```
P3= d2;
```

```
P1 = d3;
```

```
Delay();
```

```
Delay();
```

```
count=count+1;
```

```
} }
```

```
void Delay(void)
```

```
{ int j;
```

```
int i;
```

```
for(i=0;i<10;i++)
```

```
{
```

```
for(j=0;j<10000;j++)
```

```
{ } } }
```

3) Hex Counter

```
#include<reg51.h>
```

```
void Delay(void);
```

```
void main (void)
```

```
{
```

```
unsigned char count=0x00;
```

```
while(1)
```

```
{ P2 = count; // LED ON
```

```
Delay();
```

```
count= count+1;
```

```
} }
```

```
void Delay(void)
```

```
{ int j;
```

```
int i;
```

```
for(i=0;i<10;i++)
```

```
{ for(j=0;j<10000;j++)
```

```
{ } } }
```

Expt 3rd : Write a program in Embedded C to rotate Stepper Motor in Clock

and Anti-clock Wise Directions.

1) Full drive Mode

```
#include<reg51.h>

void T1M1Delay(void) //To create a delay of 200
ms using T1 Timer and M1 Mode
{
    Unsigned int x;
    for (x=0; x<20; x++)
    {
        TMOD=0x10;
        TH1=0xDB;
        TL1=0xFF;
        TR1=1;
        While (TF1==0);
        TR1=0;
        TF1=0;
    } }

void main()
{
    while(1) // To repeat infinitely
    {
        P1=0x0C; //P1 = 0000 1000 First Step
        T1M1Delay();
        P1=0x06; //P1 = 0000 0100 Second Step
        T1M1Delay();
        P1=0x03; //P1 = 0000 0010 Third Step
        T1M1Delay();
        P1=0x09; //P1 = 0000 0001 Fourth Step
        T1M1Delay();
    } }
```

2) Half Drive mode

```
#include<reg51.h>

void T1M1Delay(void) //To create a delay of 200
ms using T1 Timer and M1 Mode
{
    unsigned int x;
    for (x=0; x<20; x++)
    {
        TMOD=0x10;
        TH1=0xDB;
        TL1=0xFF;
        TR1=1;
        While (TF1==0);
        TR1=0;
        TF1=0;
    } }

void main()
{
    while (1)
    {
        P1 = 0x08; //P1 = 0000 1000 First Step
        T1M1Delay();
        P1 = 0x0C; //P1 = 0000 1100 Second Step
        T1M1Delay();
        P1 = 0x04; //P1 = 0000 0100 Third Step
        T1M1Delay();
        P1 = 0x06; //P1 = 0000 0110 Fourth Step
        T1M1Delay();
        P1 = 0x02; //P1 = 0000 0010 Fifth Step
        T1M1Delay();
        P1 = 0x03; //P1 = 0000 0011 Sixth Step
        T1M1Delay();
        P1 = 0x01; //P1 = 0000 0001 Seventh Step
        T1M1Delay();
        P1= 0x09; //P1 = 0000 1001 Eight Step
        T1M1Delay();
    } }
```

Expt 4th : **Write a program for interfacing button, LED, relay & buzzer**

```
#include<p18F4520.h>

#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF
#pragma config LVP=OFF

#define SW1 PORTDbits.RD0
#define SW2 PORTDbits.RD1
#define relay PORTDbits.RD3
#define D0 PORTDbits.RD4
#define D1 PORTDbits.RD5
#define D2 PORTDbits.RD6
#define D3 PORTDbits.RD7

void left();
void right();
void delay(unsigned int);

void main()
{
    TRISD=0x03;
    PORTDbits.RD3=0;
    while(1)
    {
        if(SW1==0&&SW2==1)
        {
            relay=0;
            left();
        }
        if(SW1==1&&SW2==0)
        {
            relay=1;
            right();
        }
    }
}
```

```
void left()
{
    relay=0;
    while(SW2==1)
    {
        D0=0;D1=0;D2=0;D3=1;
        delay(10);
        D0=0;D1=0;D2=1;D3=0;
        delay(10);
        D0=0;D1=1;D2=0;D3=0;
        delay(10);
        D0=1;D1=0;D2=0;D3=0;
        delay(10);
    } }

void right()
{
    relay=1;
    while(SW1==1)
    {
        D0=1;D1=0;D2=0;D3=0;
        delay(10);
        D0=0;D1=1;D2=0;D3=0;
        delay(10);
        D0=0;D1=0;D2=1;D3=0;
        delay(10);
        D0=0;D1=0;D2=0;D3=1;
        delay(10);
    } }

void delay(unsigned int itime)
{
    int i,j;
    for(i=0;i<itime;i++)
    for(j=0;j<1275;j++);
}
```

Expt 5th : Interfacing of LCD to PIC for Display displaying different messages

```
#include <P18f4520.h>

#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF, LVP=OFF

unsigned char text1[] = {" SKNCOE"};
unsigned char text2[] = {" PUNE"};

void delay(unsigned int value);
void lcdcmd(unsigned char value);
void lcddata(unsigned char value);
void lcdinit(void);
void lcdisplay(int row,unsigned char *str);

#define ldata PORTD
#define rs PORTCbits.RC3
#define rw PORTCbits.RC4
#define en PORTCbits.RC5

void main()
{   TRISD = 0x00;
    TRISC=0x00;
    ADCON1=0x0F;
    lcdinit();
    lcddisplay(1,text1);
    lcddisplay(2,text2);
    while(1);
} void lcddisplay(int row,unsigned char *str)
{   int k;
    if (row==1)
        lcdcmd(0x80);
    else lcdcmd(0xC0);
    for(k=0;k<16;k++)
    {   if(str[k] !=0)
```

```
        lcddata(str[k]);
        else break;
    }   while((k<16)){
        lcddata(' ');
        k++; }   }

void delay(unsigned int value)
{   int i,j;
    for(i=0;i<=value;i++)
        for(j=0;j<=50;j++);
}   void lcdcmd (unsigned char value)
{   ldata=value;
    rs=0;
    rw=0;
    en=1;
    delay(1);
    en=0;
}   void lcddata (unsigned char value)
{   ldata=value;
    rs=1;
    rw=0;
    en=1;
    delay(1);
    en=0; }

void lcdinit(void) {
    lcdcmd(0x38);
    delay(1);
    lcdcmd(0x0E);
    delay(1);
    lcdcmd(0x01);
    delay(1);
    lcdcmd(0x06);
    delay(1); }
```

Expt 6th : Interfacing of 4X4 keypad and displaying key pressed on LCD

```
#include <P18f4520.h>

#include "lcd.h"

#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF, LVP=OFF

void delay (unsigned int itime);

#define R1 PORTBbits.RB0
#define R2 PORTBbits.RB1
#define R3 PORTBbits.RB2
#define R4 PORTBbits.RB3
#define C1 PORTBbits.RB4
#define C2 PORTBbits.RB5
#define C3 PORTBbits.RB6
#define C4 PORTBbits.RB7

unsigned char text1[]={"LOGSUN SYSTEMS"};
unsigned char text2[]={"KEY PRESSED:"};

void main(void)
{
    TRISD = 0x00;
    TRISC = 0x00;
    ADCON1=0x0f;
    TRISBbits.TRISB0=1;
    TRISBbits.TRISB1=1;
    TRISBbits.TRISB2=1;
    TRISBbits.TRISB3=1;
    TRISBbits.TRISB4=0;
    TRISBbits.TRISB5=0;
    TRISBbits.TRISB6=0;
    TRISBbits.TRISB7=0;

    lcdinit();
```

```
    lcddisplay(1,text1);
    lcddisplay(2,text2);
    while(1)
    {
        C1=0;C2=C3=C4=1;
        if(R1 == 0){lcdat ('F');} // Display 0
        if(R2 == 0){lcdat ('B');} // Display 4
        if(R3 == 0){lcdat ('7');} // Display 8
        if(R4 == 0){lcdat ('3');} // Display C
        C2=0;C1=C3=C4=1;
        if(R1 == 0){lcdat ('E');} // Display 1
        if(R2 == 0){lcdat ('A');} // Display 5
        if(R3 == 0){lcdat ('6');} // Display 9
        if(R4 == 0){lcdat ('2');} // Display D
        C3=0;C1=C2=C4=1;
        if(R1 == 0){lcdat ('D');} // Display 2
        if(R2 == 0){lcdat ('9');} // Display 6
        if(R3 == 0){lcdat ('5');} // Display A
        if(R4 == 0){lcdat ('1');} // Display E
        C4=0;C1=C2=C3=1;
        if(R1 == 0){lcdat ('C');} // Display 3
        if(R2 == 0){lcdat ('8');} // Display 7
        if(R3 == 0){lcdat ('4');} // Display B
        if(R4 == 0){lcdat ('0');} // Display F
    }
}
```

Expt 7th : Interface analog voltage 0-5V to internal ADC and display value on LCD.

```
#include<P18F4520.h>

#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF, LVP=OFF

void lcdcmd(unsigned char value);
void lcddata(unsigned char value);
void msdelay(unsigned int itime);

#define ldata PORTD
#define rs PORTBbits.RB3
#define rw PORTBbits.RB4
#define en PORTBbits.RB5

void main(void)
{
    unsigned int i, d;
    unsigned char val;
    temp[3];
    TRISD=0;
    PORTD=0;
    TRISB=0x00;
    PORTB=0;
    ADCON0 = 0X01;
    ADCON1 = 0X0E;
    ADCON2=0b10001010;

    msdelay(15);
    lcdcmd(0x38);
    msdelay(15);
    lcdcmd(0x0E);
    msdelay(15);
    lcdcmd(0x01);
    msdelay(15);

    lcdcmd(0x06);
    msdelay(15);
    while(1)
    {
        lcdcmd(0x80);
        msdelay(20);
        ADCON0bits.GO = 1;
        while
        (ADCON0bits.DONE ==1);
        temp[0]= (ADRESH & 0x0f);
        temp[1]= (ADRESL & 0xf0)>>4;
        temp[2]= (ADRESL & 0x0f);
        for(d=0; d<3; d++)
        {
            if (temp[d] < 10)
                temp[d] = temp[d]+0x30;
            else
                temp[d] = temp[d]+0x37;
            lcddata(temp[d]);
            msdelay(15);
        }
        msdelay(10);
    } }

    void lcdcmd (unsigned char value)
    {
        ldata=value;
        rs=0;
        rw=0;
        en=1;
        msdelay(1);
        en=0;
    }
}
```

```
void lccdata (unsigned char value)
```

```
{  
ldata=value;  
rs=1;  
rw=0;  
en=1;  
msdelay(1);  
en=0;  
}
```

```
void msdelay (unsigned int itime)
```

```
{  
int i,j;  
for(i=0;i<itime;i++)  
for(j=0;j<1235;j++);  
}
```

Expt 8th : Interfacing serial port with PC both side communication

```
#include <p18f4520.h>
```

```
//Configuration bit setting//
```

```
#pragma config OSC = HS //Oscillator Selection
```

```
#pragma config WDT = OFF //Disable Watchdog  
timer
```

```
#pragma config LVP = OFF //Disable Low Voltage  
Programming
```

```
#pragma config PBDEN = OFF //Disable PORTB  
Analog inputs
```

```
void Transmit_String(unsigned char *string)
```

```
{  
unsigned char i=0;  
for(i=0;string[i]!='\0';i++) //loop till end of the  
string  
{  
while(PIR1bits.TXIF == 0);  
TXREG = string[i];
```

```
} }
```

```
unsigned char MSG1[] = {"UART  
COMMUNICATION \r\n"};
```

```
unsigned char MSG2[] = {"TRANSMITTING STRING  
\r\n"};
```

```
unsigned char MSG3[] = {"SEND 10 Characters  
\r\n"};
```

```
unsigned char MSG4[] = {"Received Data \r\n"};
```

```
void main(void)
```

```
{  
unsigned char j=0;  
unsigned char RX_DATA[20];  
unsigned char MSG5[] = {"Received Data \r\n"};  
TRISCbits.TRISC7 = 1; // RXD as Input  
TRISCbits.TRISC6 = 0; // TXD as Output  
RCSTA = 0x90;  
TXSTA = 0x24;  
BAUDCON = 0x00;  
SPBRG = 0x19;  
SPBRGH = 0;  
Transmit_String (MSG1);  
Transmit_String (MSG2);  
Transmit_String (MSG3);  
for (j=0; j<10 ; j++)  
{  
while(PIR1bits.RCIF == 0);  
RX_DATA[j] = RCREG;  
}  
RX_DATA[10] = '\0';  
Transmit_String (MSG4);  
Transmit_String (RX_DATA);  
while(1);  
}
```