**MC ALL EXPERIMENTS**

**EXP 1: -**

**1.Addition :-**

ORG 0000H

SJMP 30H

ORG 30H

MOV A, #05H

MOV B, #02H

ADD A,B

MOV R0,A

END

**2.SUBB**

ORG 0000H

SJMP 30H

ORG 30H

MOV A, #05H

MOV B, #02H

SUBB A,B

MOV R0,A

END

**3.MUL**

ORG 0000H

SJMP 30H

ORG 30H

MOV A, #05H

MOV B, #02H

MUL AB

MOV R0,A

END

**4.DIV**

ORG 0000H

SJMP 30H

ORG 30H

MOV A, #06H

MOV B, #02H

DIV AB

MOV R0,A

END

**EXP N0 .2 (smallest and largest numbers)**

ORG 0000H  
SJMP 30H  
ORG 30H  
MOV R0,#30H  
MOV R2,#05H  
MOV A, @R0  
DEC R2

L3: INC R0  
MOV B,@R0  
CJNE A, B, L1  
SJMP L2

L1 : **JNC** L2  
MOV A,B

L2:DJNZ R2,L3  
MOV 38H,A  
END

**Experiment No 3 ( Ascending and Desceding Order ):**

**1.Asceding Order :**

SJMP 30H

ORG 30H

MOV R2 ,#04

OUTER:

MOV R1 ,#04

MOV R0,#30H

INNER:

MOV A,@R0

INC R0

MOV B,@R0

CJNE A,B,CHECK

SJMP NEXT

CHECK:

**JC** NEXT

MOV @R0,A

DEC R0

MOV @R0,B

INC R0

NEXT:

DJNZ R1,INNER

DJNZ R2,OUTER

END

**2.Desceding Order :**

SJMP 30H

ORG 30H

MOV R2 ,#04

OUTER:

MOV R1 ,#04

MOV R0,#30H

INNER:

MOV A,@R0

INC R0

MOV B,@R0

CJNE A,B,CHECK

SJMP NEXT

CHECK:

**JNC** NEXT

MOV @R0,A

DEC R0

MOV @R0,B

INC R0

NEXT:

DJNZ R1,INNER

DJNZ R2,OUTER

END

**EXPERIMENT NO 4 :( internal to external and Decimal to hexadecimal)**

**1.Internal To External Memory :**

ORG 0000H

MOV R0,#30H

MOV DPTR ,#1020H

MOV R2,#05

L1:

MOV A,@R0

MOVX @DPTR,A

INC R0

INC DPTR

DJNZ R2,L1

END

**2.Decimal To Hexadecimal :**

ORG 0000H

SJMP 30H

ORG 30H

MOV DPTR,#4000H

MOVX A,@DPTR

ANL A,#0F0H

SWAP A

MOV B,#0AH

MUL AB

MOV R1,A

MOVX A,@DPTR

ANL A,#0FH

ADD A,R1

INC DPTR

MOVX @DPTR,A

END

**EXPERIMENT NO 5 :(**

**1.Interface Of LED :**

#include<REG51.h>

void msdelay(unsigned int);

void main()

{

P1 = 0X00;

while(1)

{

P1 = 0XFF;

msdelay(250);

P1 = 0X00;

msdelay(250);

}

}

void msdelay(unsigned int delay)

{

int x,y;

for(x=0 ; x<delay ; x++)

for(y=0 ; y<1275 ; y++);

}

**2.LED Using Timer :-**

#include <reg51.h>  
sbit A=P1^0;  
  
void Timer0\_ISR(void) interrupt 1 {  
    P1 = ~P1;      
}  
  
void main() {

**P1=0XFF;**  
    TMOD = 0x01;     
    TH0 = 0x3C;      
    TL0 = 0xB0;  
    IE = 0x82;       
    TR0 = 1;         
    while(1);        
}

**EXPERIMENT 6 :- (Square wave)**

#include <REG51.h>

sbit A = P1^0;

void Timer0\_ISR(void) interrupt 1 {

A = ~A;

}

void main() {

TMOD = 0x01;

TH0 = 0x00;

TL0 = 0x00;

IE = 0x82;

TR0 = 1;

while (1);

}

**EXPERIMENT 7 :- (**for serial communication)

**1.SOFTWARE ;**

#include<reg51.h>

void main ()

{

SCON =0X50;

TMOD=0X20;

TH1=-3;

TR1=1;

SBUF='I';

while(TI==0);

TI=0;

}

**2.HARDWARE**

#include <reg51.h>

unsigned char receive(void)

{

unsigned char rec;

while (RI == 0);

rec = SBUF;

RI = 0;

return rec;

}

void trans(unsigned char ch)

{

SBUF = ch;

while (TI == 0);

TI = 0;

}

void main(void)

{

TMOD = 0x20;

TH1 = 0xFD;

SCON = 0x50;

TR1 = 1;

IE = 0x90;

while (1)

{

trans(receive());

}

}

**EXPERIMENT 8 :- (ADC)**

#include <reg51.h>  
#define ADCPORT P2  
#define LEDPORT P0  
  
sbit SEL\_A = P1^0;  
sbit SEL\_B = P1^1;  
sbit SEL\_C = P1^2;  
sbit ALE = P1^3;  
sbit START = P1^4;  
sbit EOC = P1^5;  
sbit OE = P1^6;  
unsigned char ADC\_read(unsigned char channel)  
{  
    unsigned char result;  
    SEL\_A = SEL\_B = SEL\_C = 0;    
    P1=P1|(channel & 0X07);  
ALE = 1;    
    START = 1;  
ALE = 0;  
    START = 0;  
    while (EOC == 1);    
    while (EOC == 0);    
    OE = 1;  
    result = ADCPORT;    
    OE = 0;  
    return result;  
}  
void main() {  
    ADCPORT = 0xFF;    
    EOC = 1;    
    ALE = 0;  
    OE = 0;    
    START = 0;    
       
    while (1) {  
        LEDPORT = ADC\_read(0);  
    }  
}

**EXPERIMENT NO 9 :- ( Triangular Wave )**

#include<REG51.h>  
unsigned int d ;  
void main()  
{  
  P1=0x00;  
  while(1)  
  {  
   for(d=0 ;d<255;d++)  
   {  
    P1=d;  
   }  
   for(d=255;d>0;d--)  
   {  
    P1=d;  
   }  
  }  
}

**EXPERIMENT NO 10 ( LCD interfacing )**

#include <reg51.h>

sbit rs = P3^2;

sbit en = P3^3;

void lcdcmd(unsigned char value);

void lcddata(unsigned char value);

void msdelay(unsigned int itime);

void main(void)

{

P0 = 0x00;

P1 = 0x00;

P2 = 0x00;

P3 = 0x00;

msdelay(1000)

lcdcmd(0x38);

msdelay(250);

lcdcmd(0x0E);

msdelay(250);

lcdcmd(0x01);

msdelay(250);

lcdcmd(0x06);

msdelay(250)

lcdcmd(0x80);

msdelay(250);

lcddata('P');

msdelay(250);

lcddata('C');

msdelay(250);

lcddata('E');

msdelay(250);

lcddata('T');

msdelay(250);

lcddata('"');

msdelay(250);

lcddata('s');

msdelay(250);

lcdcmd(0x08);

msdelay(250);

lcddata('P');

msdelay(250);

lcddata('C');

msdelay(250);

lcddata('C');

msdelay(250);

lcddata('O');

msdelay(250);

lcddata('E');

msdelay(250);

}

void lcdcmd(unsigned char value)

{

P0 = value;

rs = 0;

en = 1;

msdelay(1);

en = 0;

}

void lcddata(unsigned char value)

{

P0 = value;

rs = 1;

en = 1;

msdelay(1);

en = 0;

}

void msdelay(unsigned int itime)

{

unsigned int i, j;

for (i = 0; i <= itime; i++)

for (j = 0; j <= 120; j++);

}

**EXPERIMENT 11: ( Stepper Motor )**

#include <reg51.h>

void delay(unsigned int);

void main()

{

P1 = 0x00;

delay(1);

while (1)

{

P1 = 0x01;

delay(0.00001);

P1 = 0x02;

delay(0.00001);

P1 = 0x04;

delay(0.00001);

P1 = 0x08;

delay(0.00001);

}

}

void delay(unsigned int value)

{

unsigned int i, j;

for (i = 0; i <= value; i++)

for (j = 0; j <= 1275; j++);

}

**Experiment NO 12 ( DC MOTOR )**

#include<reg51.h> sbit MOTOR1 = P1^1; sbit MOTOR2 = P1^2;

void delay (unsigned int time)

{

unsigned int i,j;

for (i=0; i<time; i++) for (j=0; j<2000; j++);

}

void DCMOTOR\_stop(void)

{

MOTOR1 = 0;

MOTOR2 = 0;

}

void DCMOTOR\_forward(void)

{

MOTOR1 = 1;

MOTOR2 = 0;

}

void DCMOTOR\_reverse(void)

{

MOTOR1 = 0;

MOTOR2 = 1;

}

int main ()

{

P1 = 0;

while (1)

{

DCMOTOR\_stop();

delay(100);

DCMOTOR\_forward();

delay(100);

DCMOTOR\_stop();

delay(100);

DCMOTOR\_reverse();

delay(100);

}

**EXPERIMENT NO 13 : ( switch press LED )**

#include <reg51.h>

#define SWITCH\_PORT P3

#define LED\_PORT P2

int main()

{

P3 = 0xFF;

P2 = 0x00;

while(1)

{ if((P3&0x08)==0x08)

{

P2 = 0x92;

}

else

{

P2 = 0x6D;

}

}

}

**EXPERIMENT 14 :-**