

//PRactical 1 Internal to internal:

```
ORG 0000H
MOV R7,#05H
MOV R0,#30H
MOV R1,#50H
LOOP: MOV A,@R0
MOV @R1,A
INC R0
INC R1
DJNZ R7, LOOP
NOP
END
```

//• Internal to external

```
ORG 0000H
MOV R2,#0AH
MOV R0,#50H
MOV DPTR,#60h
LOOP: MOV A,@R0
MOVX @DPTR,A
INC R0
INC DPTR
DJNZ R2, LOOP
NOP
END
```

case1:// Program for Toggle

//All bits of Port

#include<reg51.h>

void Delay(void);

void main (void)

{ while(1) // infinite loop

{ P3 = 0x00; // LED ON

Delay();

P3 = 0xff; // LED OFF

Delay(); } }

void Delay(void) {

int i, j;

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

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

//case2:

#include<reg51.h>

void T1M1Delay(void);

void main (void) {

unsigned char x;

P2=0x55;

while(1) // infinite loop

{ P2=~P2;

//toggle all bits of P2

P3 = 0x00;

for(x=0; x<20;x++)

T1M1Delay(); } }

void T1M1Delay(void) {

TMOD= 0x10;

// TIMER 1 MODE 1 (16 BIT)

TL1= 0xFE; //LOAD TL1

TH1=0xA5; //LOAD TH1

TR1=1; // START TIMER 1

while(TF1==0);

//WAIT TILL TF1 TO ROLL OVER

TR1=0; //STOP TIMER

TF1=0; // RESET FLAG }

//case3:hex counter

#include <reg51.h>

void delay(){

unsigned int i, j;

for(i = 0; i < 255; i++){

for(j = 0; j < 255; j++){

// Empty loop for delay}

}}

void main() {

unsigned int z;

while(1) { // Infinite loop

for(z = 0x00; z<=0xff; z++)

{ P1 = z; // Output count to

Port 1 //(LEDs)

delay();

// Delay for visible change

}}

PR:3 7seg

#include <reg51.h>

#define SEGMENT_PORT P2

//Define the port connected to the

//7-segment display

// Segment values for digits 0 to 9

unsigned char segment_code[10] =
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x
7D,0x07,0x7F,0x6F};

void delay(unsigned int ms){

unsigned int i, j;

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

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

}}

void main() {unsigned char i;

while (1){

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

SEGMENT_PORT =

segment_code[i]; // Send the

//segment code to the port

delay(1000); // Delay for 1 second

}}

// Segment codes for displaying

//numbers 0 to 9 on a common

//anode display

// 0 = 0xC0, 1 = 0xF9, 2 = 0xA4, 3 =

//0xB0, 4 = 0x99, 5 = 0x92, 6 =

//0x82, 7 = 0xF8, 8 = 0x80, 9 =0x90

unsigned char segment_code[] =

{0xC0, 0xF9, 0xA4, 0xB0, 0x99,

0x92, 0x82, 0xF8, 0x80, 0x90};

void delay(unsigned int ms) {

unsigned int i, j;

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

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

}void main() {

unsigned char i;

while(1) { for(i = 0; i < 10; i++) {

P2 = segment_code[i]; // Output

//the segment code to PORT2

delay(500); // Delayof 500ms

}} }

//PR4 fulldrive stepper

#include<reg51.h>

void msdelay(unsigned int

time)

{ unsigned i,j ;

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

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

void main(){

while(1) {

P2 = 0x03;

//0011P2_0=1,P2_1=1,

//P2_2 =0,P2_3=0

msdelay(1);

P2 = 0x06; //0110

msdelay(1);

P2 = 0x0C; //1100

msdelay(1);

P2 = 0x09; //1001

msdelay(1); } }

```

//halfdrivestep:
#include<reg51.h>
void msdelay(unsigned int
time)
{ unsigned i,j ;
for(i=0;i<time;i++)
for(j=0;j<1275;j++); }
void main() {
while(1) {
P2 = 0x08; //1000
msdelay(1);
P2 = 0x0c; //1100
msdelay(1);
P2 = 0x04; //0100
msdelay(1);
P2 = 0x06; //0110
msdelay(1);
p2 = 0x02; //0010
msdelay(1);
p2 = 0x03; //0011
msdelay(1);
p2 = 0x01; //0001
msdelay(1);
p2 = 0x09; //1001
msdelay(1); } }

```

//Experiment no. 7

```

//Aim: To generate square wave
//using timer of PIC18F with
#include<P18F4520.h>
void delay_timer() {
    TOCON = 0x08;
// Timer 0, 16-bit mode, no
//prescaler
    TMR0H = 0xFB;
// Value for 0.1 //ms delay
    TMR0L = 0x50;
    TOCONbits.TMR0ON = 1;
//Timer 0 On
    while (INTCONbits.TMR0IF == 0);
// Wait for Timer 0 interrupt flag
    TOCONbits.TMR0ON = 0;
// Timer 0 Off
    INTCONbits.TMR0IF = 0;
// Clear Timer 0 interrupt flag
} void main() {
    TRISCbits.TRISC0 = 0;

```

```

// Configure RC0 as output
// (onPORTC)
while (1) {
    PORTCbits.RC0 = 1;
// Set RC0 high
    delay_timer();
// Call delay function
    PORTCbits.RC0 = 0;
// Set RC0 low
    delay_timer();
// Call delay function } }

```

//Experiment no. 8

```

//Write a C program for the PIC18
//to transfer the letter 'G' serially
//at 9600 baud,
#include <P18F4520.h>
void main(void){
    TXSTA=0x20;
//choose low baud rate, 8-bit
    SPBRG=15;
//9600 baud rate/ XTAL 10 MHz
    TXSTAbits.TXEN=1;
// Enable transmission
    RCSTAbits.SPEN=1;
// Enable serial port (configures
//TX/CK pins)
    while (1) {
        TXREG='G'; //place value in buffer
        while (PIR1bits.TXIF==0);
//wait until all gone
    } }

```

//Experiment No 9:

```

//DC Motor control and PWM
// usingC.
# include<p18f4520.h>
void main(){
    TRISC = 0xFB;
// make CCP1 output Pin
    TRISD = 0x80;
// make RD7 input Pin
    CCP1CON = 0x3C;
//PWM Mode ,11 for DC1B1:B0
    PR2 =100;
// set period to 100*16/Fosc
    T2CON = 0x01;
//4 Prescaler, no postscaler
    while (1) {
        if(PORTDbits.RD7==1)
            CCPR1L=25;
//25% duty Cycle

```

```

else
    CCPR1L=50; //50% duty Cycle
    TMR2=0x0; // clear timer 2
    PIR1bits.TMR2IF=0;
// Clear timer2 Flag
    T2CONbits.TMR2ON=1;
// Start Timer 2
    while(PIR1bits.TMR2IF==0);
//wait for end of period
    } }

```

//Interfacing LCD

```

1=LMO16L
2=DC
3=POT
4=power

```

//Squire wave

```

1=virtual instrumr=ention mode=
Oscilloscope

```

//interfacing serial port PC

```

1=terminal

```

//DC motor

```

1=MOTOR
2=L293D
3= 3 power

```

//7 segment display

```

1=7 senment

```

//stepper motor

```

1=ULN2003A
2=stepper moter

```

```

// (Relay1=1, Relay2=1) & (LED's
//sequence Left to Right=0111=7)
Delay10KTCYx(100); // 400mSDelay
PORTD = 0x3B;
// (LED's sequence Left to
// Right=1011=B)
Delay10KTCYx(100);
PORTD = 0x3D;
Delay10KTCYx(100);
PORTD = 0x3E;
Delay10KTCYx(100);
if(!SWITCH0)
// check if 2nd switch is pressed
break; } }
else if(!SWITCH0)
// Condition for 2nd switch {
while (1) {
BUZZER =0; // Buzzer Off
PORTD = 0xcE;
// (Relay1=0,Relay2=0) & (LED's
//sequence Right to Left=1110=E)
Delay10KTCYx(100);
PORTD = 0xcD;
// LED's sequence Right to
//Left=1101=D
Delay10KTCYx(100);
PORTD = 0xcB;
Delay10KTCYx(100);
PORTD = 0xc7;
Delay10KTCYx(100);
if(!SWITCH1)
// check if 1st switch is pressed
break;
} } } }

```

//Expt.5: Interfacing LEDs, Switches, Buzzer and Relay

```

#include <p18f4520.h>
//Include Controller specific .h
#include <delays.h>
#define BUZZER PORTAbits.RA3
//Buzzer connected to PORTA 3rd
#define SWITCH0 PORTBbits.RB0
//Switch0 connected to PORTB 0th
#define SWITCH1 PORTBbits.RB1
//Switch1 connected to PORTB 1st

```

```

void main(void) {
TRISA = 0x00;
// RA3, Output Direction
TRISB = 0xff;
// RB0, B1 Input Direction
TRISD = 0x00; // [RD0-3=LED's]
//[RD4,5=Relay1,2] Output
Direction
PORTD = 0xff; // [RD0-3=LED's]
//[RD4,5=Relay1,2] Initialise as 0xff
while (1) {
if(!SWITCH1) // Condition for 1st
switch { while (1) {
BUZZER =1; // Buzzer On
PORTD = 0x37;

```

PR 6: LCD

```

#include<p18f4520.h>
#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF,
LVP=OFF
//Function Prototype declaration
void lcdcmd (unsigned char value) ;
void lcddata (unsigned char value);
void msdelay(unsigned int itime);
#define ldata PORTD
//Declare ldata variable for PORTD
#define rs PORTEbits.RE0
//Declare rs variable for pin RE0

```

```

#define rw PORTEbits.RE1
//Declare rw variable for pin RE1
#define en PORTEbits.RE2
//Declare en variable for pin RE2
void main(){
TRISD = 0x00;
//Set direction of PORTD as output
TRISE=0X00;
//set direction of PORTE as output
msdelay(50);
lcdcmd(0x38); //16x2 LCD
msdelay(50);
lcdcmd(0x0E);
// Display on Cursor on
msdelay(15);
lcdcmd(0x01);
//clear Display screen
msdelay(15);
lcdcmd(0x06);
//Increment cursor and shift right
msdelay(15);
lcdcmd(0x80); //Force cursor on
//first row first position
lcddata('S'); //Display character 'S'
msdelay(50);
lcddata('P'); //Display character 'P'
msdelay(50);
lcddata('P'); //Display character 'P'
msdelay(50);
lcddata('U'); //Display character 'U'
msdelay(50);
lcdcmd(0xC0); //NEXT LINE
msdelay(15);
lcddata('S'); //Display character 'S'
msdelay(50);
lcddata('I'); //Display character 'P'
msdelay(50);
lcddata('T'); //Display character 'P'
msdelay(50);
lcddata('S'); //Display character 'U'
msdelay(50);
}
void lcdcmd (unsigned char value)
{
ldata=value;
//Send the command value to
//PORTD
rs=0; //selection of command
//register of LCD
rw=0; // write mode is selected
en=1; //Generate High to Low
//pulse on Enable pin

```

```
msdelay(1);
en=0; }
void lcddata (unsigned char value)
{ ldata=value;
//Send the command value to
//PORTD
rs=1;
//selection of DATA register of LCD
rw=0; // write mode is selected
en=1; //Generate High to Low
//pulse on Enable pin
msdelay(1);
en=0; }
void msdelay (unsigned int itime) {
int i,j;
for(i=0;i<itime;i++){
for(j=0;j<135;j++); } }
```