**Experiment no 1**

**C program to add two 8-bit numbers**

#include<reg51.h>

void main(void)

{

unsigned char x,y,z;

x=0x0A;

y=0x05;

P1=0x00;

z=x+y;

P1=z;

}

**C program to subtract two 8-bit numbers**

#include<reg51.h>

void main(void)

{

unsigned char x,y,z;

x=0x08;

y=0x04;

P1=0x00;

z=x-y;

P1=z;

}

**C program to multiply two 8-bit numbers**

#include&lt;reg51.h&gt;

void main(void)

{

unsigned char x,y,z;

x=0x10;

y=0x02;

P1=0x00;

z=x\*y;

P1=z;

}

**C program to divide two 8-bit numbers**

#include<reg51.h>

void main(void)

{

unsigned char x,y,z;

x=0x10;

y=0x02;

P1=0x00;

z=x/y;

P1=z;

}

**C program to divide two 8-bit numbers**

#include<reg51.h>

void main(void)

{

unsigned char mynum[]= “012345ABCD”

unsigned char z;

for(z=0; z&lt;=9; z++)

P1= mynum[z];

}

**Experiment No: 2**

* **LED Flashing**

#include<reg51.h>

void Delay(unsigned int time);

void main()

{P1=0x00;

while(1)

{

P1=0x0FF;

Delay(250);

P1=0x00;

Delay(250);

}}

void Delay(unsigned int time)

{

int i,j;

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

for(j=0;j<165;j++);

}

* **LED Toggling**

#include<reg51.h>

void Delay(unsigned int time);

void main()

{

while(1)

{

P1=0x0AA;

Delay(250);

P1=~P1;

Delay(250);

}

}

void Delay(unsigned int time)

{

int i,j;

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

for(j=0;j<165;j++);

}

**Experiment 3**

* **Square Wave:**

#include<Reg51.h>

void delay(unsigned int time);

void main()

{

while(1)

{

P0=0XFF;

delay(25);

P0=0X00;

delay(25);

}

}

void delay(unsigned int time)

{

int i,j;

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

for(j=0;j<5000;j++);

**}**

* **Triangular wave :**

#include<reg51.h>

void main()

{int count;

P0=0X00;

P0=0X00;

for(count=0X00;count<0Xff;count++)

{P0=count;

}

P0=0Xff;

P0=0Xff;

for(count=0Xff;count<0X00;count--)

{P0=count;

}

}

**Experiment No 4**

**Stepper Motor**

#include<reg51.h>

void delay(unsigned int time);

void main() {

P0 = 0x00;

while (1) {

P0 = 0x09;

delay(100);

P0 = 0x0A;

delay(100);

P0 = 0x06;

delay(100);

P0 = 0x05;

delay(100);

}

}

void delay(unsigned int time) {

unsigned int i;

TMOD = 0x10;

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

TH1 = 0xFF;

TL1 = 0x49;

TR1 = 1;

while (TF1 == 0);

TR1 = 0;

TF1 = 0;

}

}

**Experiment no : 5**

LED Buzzer

#include <p18f4550.h>

#pragma config FOSC=HS

#pragma config WDT=OFF

#pragma config LVP=OFF

#pragma config PBADEN=OFF

#define lrbit PORTBbits.RB4 //SW0 interfaced to RB4

#define rlbit PORTBbits.RB5 //SW1

#define buzzer PORTCbits.RC2

#define relay PORTCbits.RC1

void MsDelay (unsigned int time)

{

unsigned int i, j;

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

for (j = 0; j < 275; j++);

}

void main()

{

unsigned char val=0;

INTCON2bits.RBPU=0;

ADCON1 = 0x0F;

TRISBbits.TRISB4=1;

TRISBbits.TRISB5=1;

TRISCbits.TRISC1 = 0;

TRISCbits.TRISC2 = 0;

TRISD = 0x00;

PORTD = 0x00;

buzzer = 0;

relay=0;

while (1)

{ if (!(lrbit)) // if (lrbit == 0)

val = 1;

if(!(rlbit))

val = 2;

if (val == 1)

{

buzzer = 1;

relay = 1;

PORTD = PORTD >>1;

if (PORTD == 0x00)

PORTD = 0x80;

}

MsDelay(250);

}

if (val == 2)

{

buzzer = 0;

relay = 0;

PORTD = PORTD<<1;

if (PORTD== 0x00)

PORTD = 0x01;

MsDelay(250);

}

}

**Experiment no 6**

LCD with PIC

#include <p18f4550.h>

// Configuration Bits

#pragma config FOSC = HS

#pragma config WDT = OFF

#pragma config LVP = OFF

#pragma config PBADEN = OFF

#define RS PORTCbits.RC0

#define RW PORTCbits.RC1

#define EN PORTCbits.RC2

void LCD\_data(unsigned char data);

void LCD\_cmd(unsigned char cmd);

void LCD\_init();

void delay(unsigned int time);

void main() {

TRISD = 0x00; // Set PORTD as output for data

TRISC = 0x00; // Set PORTE as output for control pins

LCD\_init();

LCD\_cmd(0x80); // Move cursor to the beginning of the first line

LCD\_data('L');

delay(25);

LCD\_data('C');

delay(25);

LCD\_data('D');

delay(25);

LCD\_data('P');

delay(25);

LCD\_data('I');

delay(25);

LCD\_data('C');

delay(25);

LCD\_data('1');

delay(25);

LCD\_data('8');

while (1); // Keep looping to prevent the program from exiting

}

void LCD\_data(unsigned char data) {

PORTD = data; // Send data to the data port

RS = 1; // Set RS to 1 for data

RW = 0; // Set RW to 0 for write

EN = 1; // Enable LCD

delay(25);

EN = 0; // Disable LCD

}

void LCD\_cmd(unsigned char cmd) {

PORTD = cmd; // Send command to the data port

RS = 0; // Set RS to 0 for command

RW = 0; // Set RW to 0 for write

EN = 1; // Enable LCD

delay(25);

EN = 0; // Disable LCD

}

void LCD\_init() {

delay(15); // Wait for LCD to power up

LCD\_cmd(0x38); // 8-bit mode, 2 lines, 5x7 matrix

delay(5);

LCD\_cmd(0x01); // Clear display

delay(2);

LCD\_cmd(0x06); // Increment cursor

delay(1);

LCD\_cmd(0x0C); // Display on, cursor off

delay(1);

LCD\_cmd(0x80); // Move cursor to the beginning of the first line

delay(1);

}

void delay(unsigned int time) {

int i, j;

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

for (j = 0; j < 275; j++); // Simple delay loop

}

**Experiment no 7**

**LCD with 8051 in proteus**

#include <reg51.h> // Include register definition file for AT89C51

#define LCD P0 // Define LCD data port

sbit RS = P1^0; // Register Select pin

sbit RW = P1^1; // Read/Write pin

sbit E = P1^2; // Enable pin

void Delay(unsigned int ms);

void LCD\_Command(unsigned char cmd);

void LCD\_Char(unsigned char Data);

void LCD\_Init(void);

void LCD\_String(char \*str);

void main() {

LCD\_Init(); // Initialize LCD

LCD\_String("ISQUAREIT"); // Display message

while (1); // Loop forever

}

void LCD\_Init(void) {

Delay(20); // Wait for more than 15 ms after power on

LCD\_Command(0x38); // 2 Lines, 5x7 Matrix

Delay(5); // Wait for a bit after this command

LCD\_Command(0x0C); // Display ON, Cursor OFF

LCD\_Command(0x06); // Increment cursor

LCD\_Command(0x01); // Clear display

Delay(2); // Wait for clear command to complete

}

void LCD\_Command(unsigned char cmd) {

RS = 0; // Command mode

RW = 0; // Write mode

LCD = cmd; // Send command

E = 1; // Enable high

Delay(1); // Delay for processing

E = 0; // Enable low

Delay(5); // Wait for command to be executed

}

void LCD\_Char(unsigned char Data) {

RS = 1; // Data mode

RW = 0; // Write mode

LCD = Data; // Send data

E = 1; // Enable high

Delay(1); // Delay for processing

E = 0; // Enable low

Delay(5); // Wait for data to be written

}

void LCD\_String(char \*str) {

while (\*str) {

LCD\_Char(\*str++); // Send each character

}}

void Delay(unsigned int ms) {

unsigned int i, j;

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

for (j = 0; j < 1275; j++); // Approx delay

}}

**Experiment no 8**

**7 segment with proteus**

#include <reg51.h>

void delay (int k);

void main()

{

unsigned char i;

unsigned char arr[10] = {0x3F, 0x06, 0x5B, 0x4F, 0X66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};

P0 = 0x00;

while (1)

{

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

{

P0 = arr[i];

delay(100);

}

}

}

void delay (int k)

{

int i, j;

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

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

}

**Experiment no 9**

**Dc motor with proteus**

#include <reg51.h>

// Define motor control pins

sbit Motor\_Pin1 = P1^0; // IN1 of L293D (for motor control)

sbit Motor\_Pin2 = P1^1; // IN2 of L293D (for motor control)

sbit Button = P2^0; // Button connected to P2.0

void delay(unsigned int time) {

unsigned int i, j;

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

for (j = 0; j < 1275; j++); // Adjust for desired delay

}

}void main() {

Motor\_Pin1 = 0;

Motor\_Pin2 = 0;

while (1){

if (Button == 1) { // If the button is pressed

Motor\_Pin1 = 0; // IN1 low

Motor\_Pin2 = 1; // IN2 high (motor rotates anticlockwise)

} else { // If the button is not pressed

Motor\_Pin1 = 1; // IN1 high

Motor\_Pin2 = 0; // IN2 low (motor rotates clockwise)

}

delay(100); // Add a short delay for stability

}

}