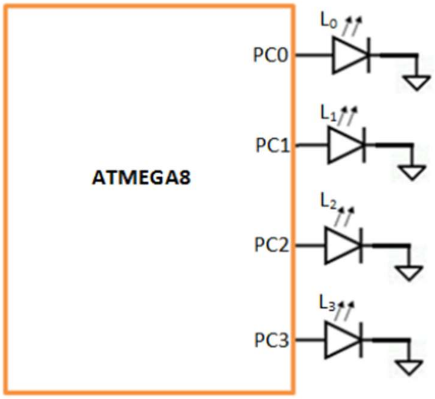
| S. No | Name of Experiment |
|---|---|
| 1. | **LED BLINKING: -** Program to cause continuous blinking of LEDs (i)simultaneously(allotonic), (ii) sequentially |
| 2. | **LED BLINKING CONTROLLED BY A SWITCH: -** (i) Program to glow an LED when its corresponding switch is pressed (ii) Program to glow L0 when SW1 is pressed, L0 and L1 when SW2 is presses, L0, L1, L2 when SW3 is presses and all the LEDs when SW4 is presses |
| 3. | **INTERFACING OF SEVEN SEGMENT DISPLAY: -** Program to display digits 0 to 9 on seven segment display. |
| 4. | **INTERFACING OF BUZZER: -**<br>Program to control buzzer |
| 5. | **BUZZER CONTROLLED BY A SWITCH: -**<br>Program to cause the buzzer to beep (i) once when switch1 (SW1) is pressed (ii) twice when switch2 (SW2) is pressed (iii) thrice when switch (SW3) is pressed and (iv) four times when swich4 (SW4) is pressed |
| 6. | **D.C. MOTOR INTERFACING: -**<br>Program to rotate a d.c. motor in clockwise and anti-clockwise direction |
| 7. | **Stepping Motor: -**<br>I. Single phase stepping motor<br>**II.** 2 phase stepping motor<br>**III.** Half phase stepping motor |

# PROGRAM-1

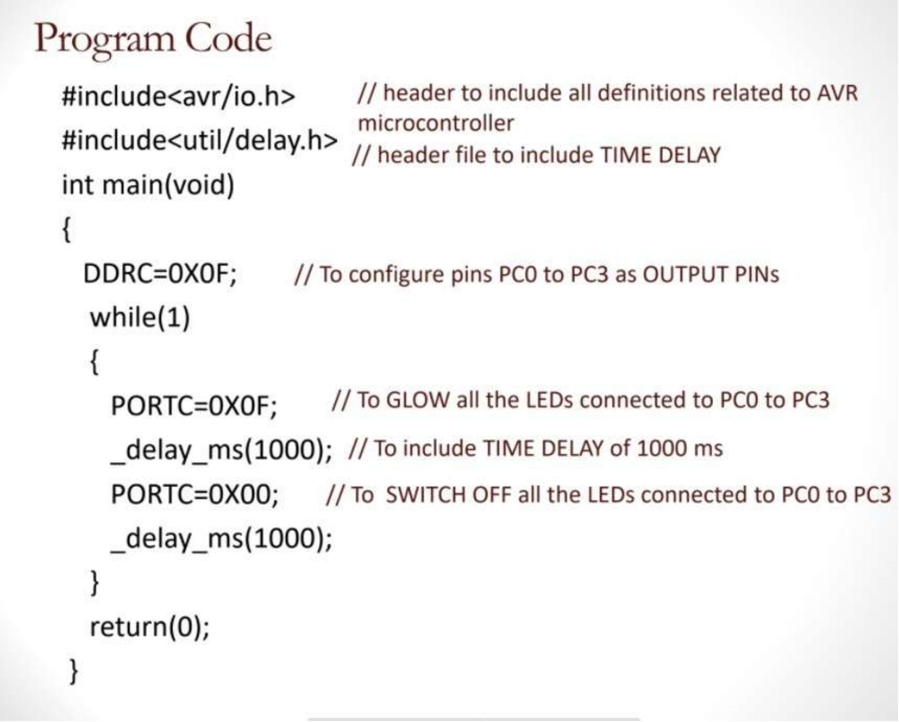
## AIM:

LEDBLINKING Program to cause continuous blinking of LEDs

(i)simultaneously(allotonic),(ii)sequentially

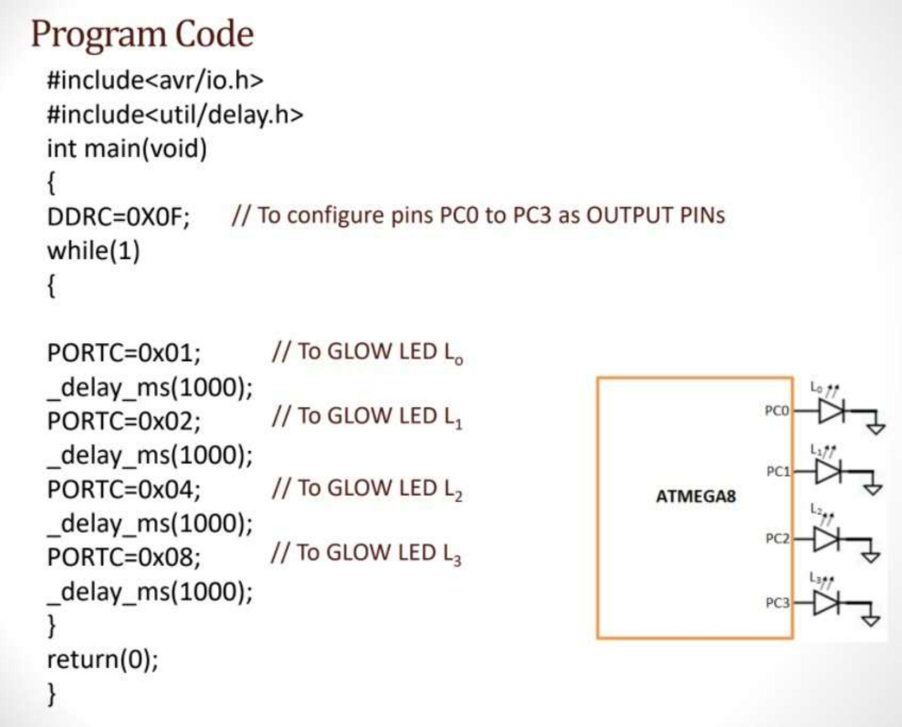


## SOURCE CODE: -



Program Code

```
#include<avr/io.h>        // header to include all definitions related to AVR
                           microcontroller
#include<util/delay.h>    // header file to include TIME DELAY
int main(void)
{
  DDRC=0X0F;       // To configure pins PC0 to PC3 as OUTPUT PINs
   while(1)
   {
     PORTC=0X0F;       // To GLOW all the LEDs connected to PC0 to PC3
     _delay_ms(1000);  // To include TIME DELAY of 1000 ms
     PORTC=0X00;       // To  SWITCH OFF all the LEDs connected to PC0 to PC3
     _delay_ms(1000);
   }
   return(0);
 }
```



## Program Code

```
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
DDRC=0X0F;     // To configure pins PC0 to PC3 as OUTPUT PINs
while(1)
{

PORTC=0x01;       // To GLOW LED Lo
_delay_ms(1000);
PORTC=0x02;       // To GLOW LED L1
_delay_ms(1000);
PORTC=0x04;       // To GLOW LED L2
_delay_ms(1000);
PORTC=0x08;       // To GLOW LED L3
_delay_ms(1000);
}
return(0);
}
```

ATMEGA8
PC0
PC1
PC2
PC3
$L_0$
$L_1$
$L_2$
$L_3$

## CONCLUSION: -

Blinking of LED’s simultaneously and sequentially has been successfully observed on AVR kit.

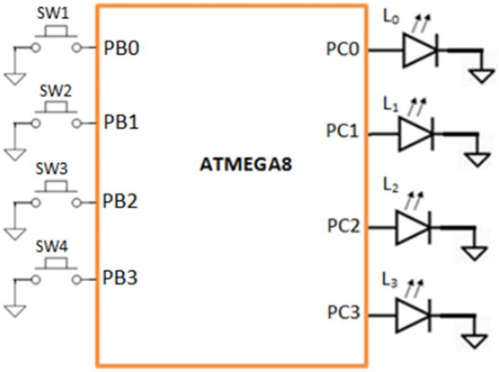
# PROGRAM – 2

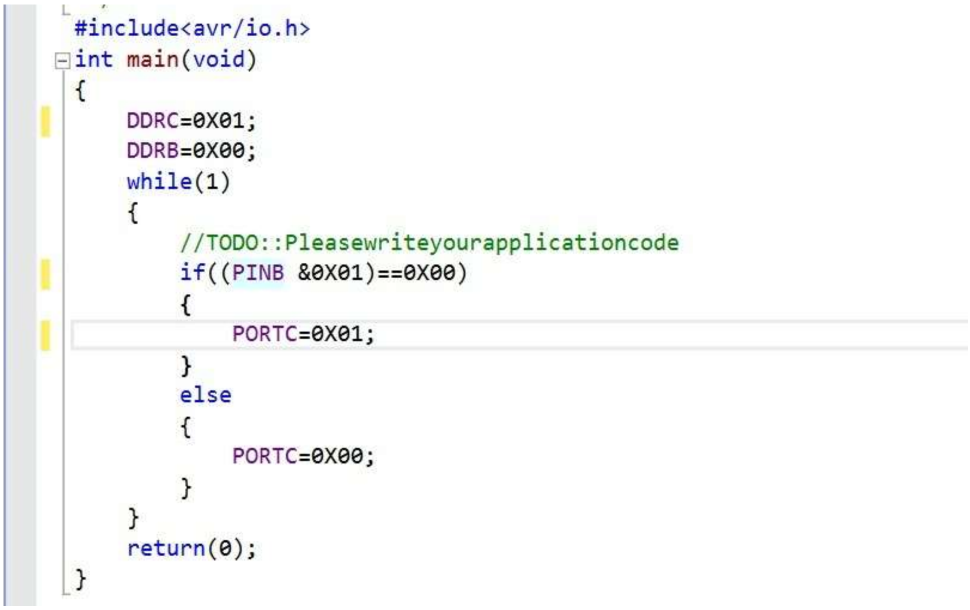
## AIM:-

LED BLINKING CONTROLLED BY A SWITCH

(i) Program to glow an LED when its corresponding switch is pressed

(ii) Program to glow L0 when SW1 is pressed, L0 and L1 when SW2 is presses, L0, L1, L2 when SW3 is presses and all the LEDs when SW4 is presses

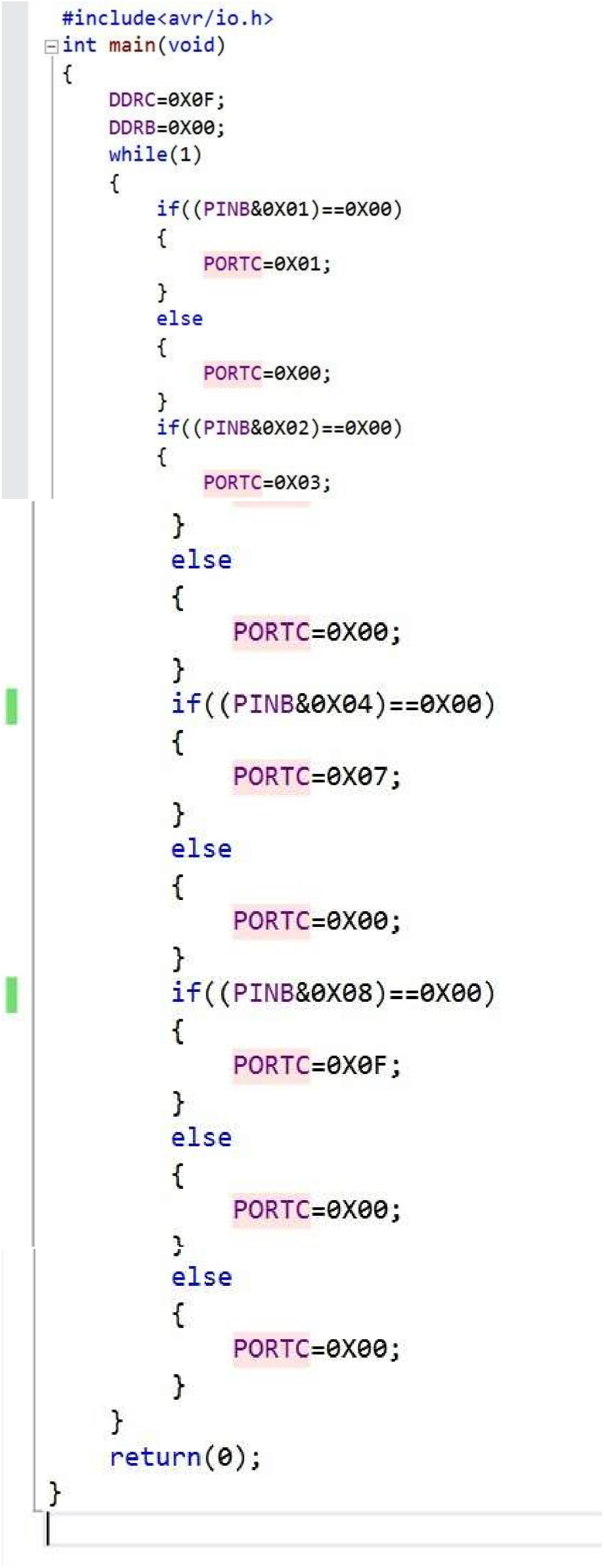




```
#include<avr/io.h>
int main(void)
{
    DDRC=0X01;
    DDRB=0X00;
    while(1)
    {
        //TODO::Pleasewriteyourapplicationcode
        if((PINB &0X01)==0X00)
        {
            PORTC=0X01;
        }
        else
        {
            PORTC=0X00;
        }
    }
    return(0);
}
```

i)

ii)



```
#include<avr/io.h>
int main(void)
{
    DDRC=0X0F;
    DDRB=0X00;
    while(1)
    {
        if((PINB&0X01)==0X00)
        {
            PORTC=0X01;
        }
        else
        {
            PORTC=0X00;
        }
        if((PINB&0X02)==0X00)
        {
            PORTC=0X03;
        }
        else
        {
            PORTC=0X00;
        }
        if((PINB&0X04)==0X00)
        {
            PORTC=0X07;
        }
        else
        {
            PORTC=0X00;
        }
        if((PINB&0X08)==0X00)
        {
            PORTC=0X0F;
        }
        else
        {
            PORTC=0X00;
        }
        else
        {
            PORTC=0X00;
        }
    }
    return(0);
}
```

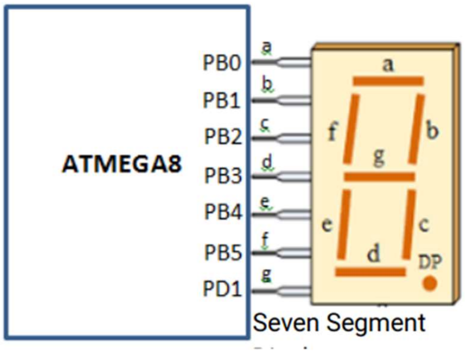
## CONCLUSION: -

The blinking of LEDs was seen by keeping the one switch on in which first L0 glows the L0 and L1 both and continues like this.

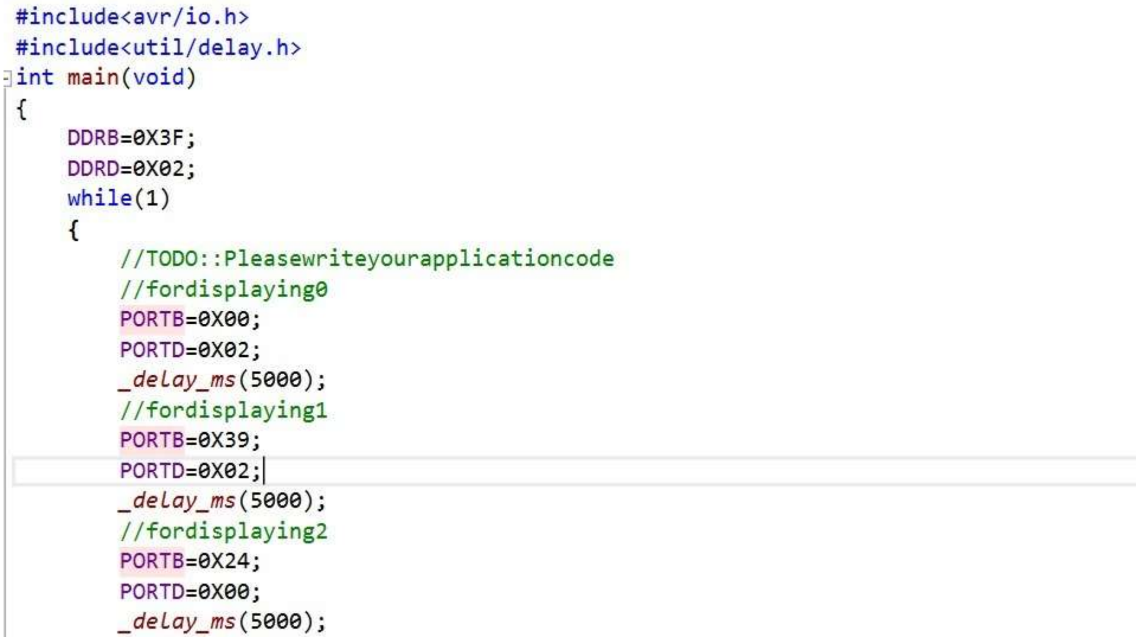
# PROGRAM – 3

## AIM: -

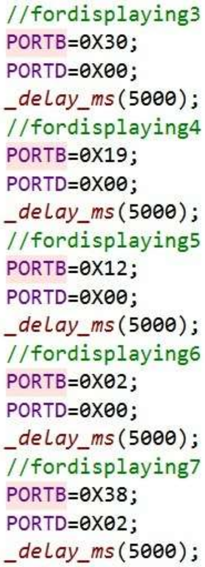
INTERFACING OF SEVEN SEGMENT DISPLAY Program to display digits 0 to 9 on seven segment display.



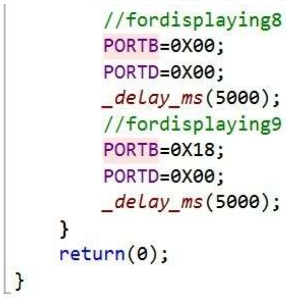
## SOURCE CODE:



```
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
    DDRB=0X3F;
    DDRD=0X02;
    while(1)
    {
        //TODO::Pleasewriteyourapplicationcode
        //fordisplaying0
        PORTB=0X00;
        PORTD=0X02;
        _delay_ms(5000);
        //fordisplaying1
        PORTB=0X39;
        PORTD=0X02;
        _delay_ms(5000);
        //fordisplaying2
        PORTB=0X24;
        PORTD=0X00;
        _delay_ms(5000);
```



```
    //fordisplaying3
    PORTB=0X30;
    PORTD=0X00;
    _delay_ms(5000);
    //fordisplaying4
    PORTB=0X19;
    PORTD=0X00;
    _delay_ms(5000);
    //fordisplaying5
    PORTB=0X12;
    PORTD=0X00;
    _delay_ms(5000);
    //fordisplaying6
    PORTB=0X02;
    PORTD=0X00;
    _delay_ms(5000);
    //fordisplaying7
    PORTB=0X38;
    PORTD=0X02;
    _delay_ms(5000);
```



```
        //fordisplaying8
        PORTB=0X00;
        PORTD=0X00;
        _delay_ms(5000);
        //fordisplaying9
        PORTB=0X18;
        PORTD=0X00;
        _delay_ms(5000);
    }
    return(0);
}
```

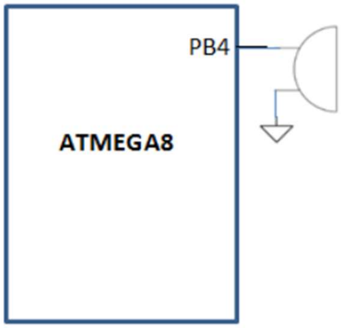
## CONCLUSION: -

All 0 to 9 digits were displayed on seven segment display drivers using ATMEGA8 board

# PROGRAM – 4

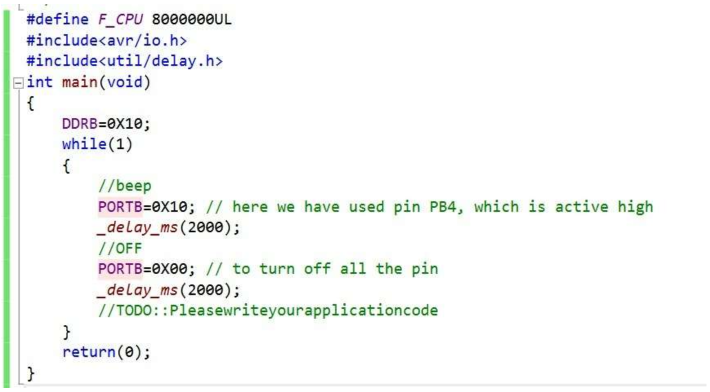
## AIM: -

Program to control buzzer (INTERFACING OF BUZZER)



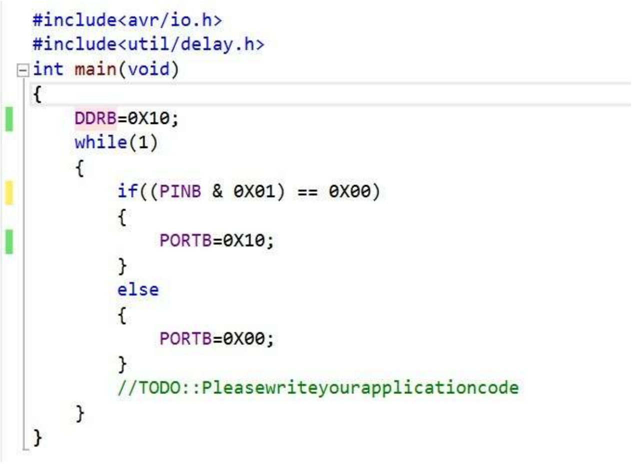
## SOURCE CODE:

### i) Simple Program to beep the buzzer



```
#define F_CPU 8000000UL
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
    DDRB=0X10;
    while(1)
    {
        //beep
        PORTB=0X10; // here we have used pin PB4, which is active high
        _delay_ms(2000);
        //OFF
        PORTB=0X00; // to turn off all the pin
        _delay_ms(2000);
        //TODO::Pleasewriteyourapplicationcode
    }
    return(0);
}
```

### ii) Program to control buzzer using switch



```
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
    DDRB=0X10;
    while(1)
    {
        if((PINB & 0X01) == 0X00)
        {
            PORTB=0X10;
        }
        else
        {
            PORTB=0X00;
        }
        //TODO::Pleasewriteyourapplicationcode
    }
}
```

## CONCLUSION: -

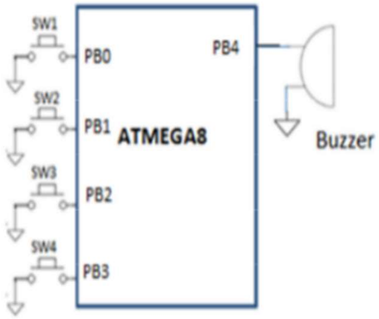
An interfacing of buzzer in ATMEGA8 board by switch and other simple mode is observed.

# PROGRAM – 5

## AIM: -

BUZZER CONTROLLED BY A SWITCH Program to cause the buzzer to beep

(i) once when switch1(SW1) is pressed
(ii) twice when switch2(SW2) is pressed
(iii) thrice when switch (SW3) is pressed and
(iv) four times when swich4 (SW4) is pressed.



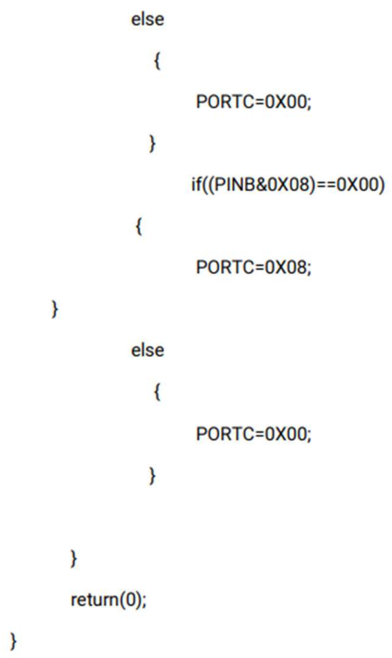
## SOURCE CODE: -

```
#include <avr/io.h>


int main(void)
```

```
{
        DDRC=0X0F;
        DDRB=0X00;
while(1)
  {
     //TODO:: Please write your application code
                if((PINB&0X01)==0X00)
                {
                        PORTC=0X01;
      }
                else
                  {
                        PORTC=0X00;
                  }
                if((PINB&0X02)==0X00)
                {
                        PORTC=0X02;
      }
                else
                  {
                        PORTC=0X00;

                }
                      if((PINB&0X04)==0X00)
               {
                       PORTC=0X04;
    }
```



```
                    else
                      {
                            PORTC=0X00;
                      }
                            if((PINB&0X08)==0X00)
                    {
                            PORTC=0X08;
            }
                    else
                      {
                            PORTC=0X00;
                      }

              }
              return(0);
        }
```

## CONCLUSION: -

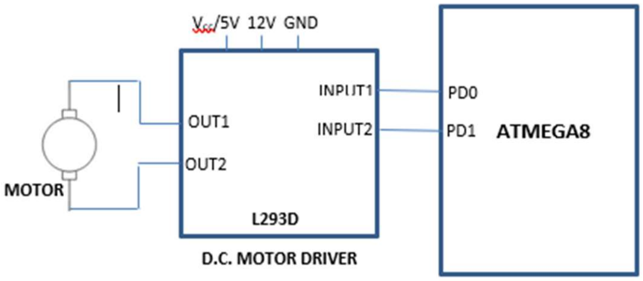
A buzzer function was seen by beeping it by twice and once in ATMEGA8 board

# PROGRAM – 6

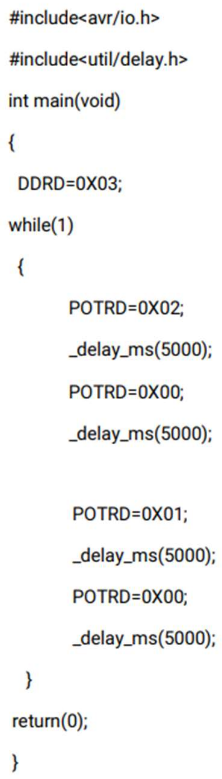
## AIM: -

**D.C. MOTOR INTERFACING**

Program to rotate a d.c. motor in clockwise and anti-clockwise direction



## SOURCE CODE: -



```
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
  DDRD=0X03;
while(1)
 {
        POTRD=0X02;
        _delay_ms(5000);
        POTRD=0X00;
        _delay_ms(5000);

         POTRD=0X01;
         _delay_ms(5000);
         POTRD=0X00;
         _delay_ms(5000);
   }
 return(0);
 }
```

## CONCLUSION: -

A motor is interfaced to rotate in clockwise and anticlockwise direction in ATMEGA8 board

# PROGRAM – 7

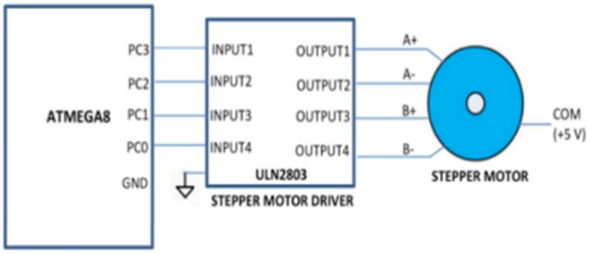
## AIM: -

**Stepping motor interfacing**

1. Single phase stepping motor

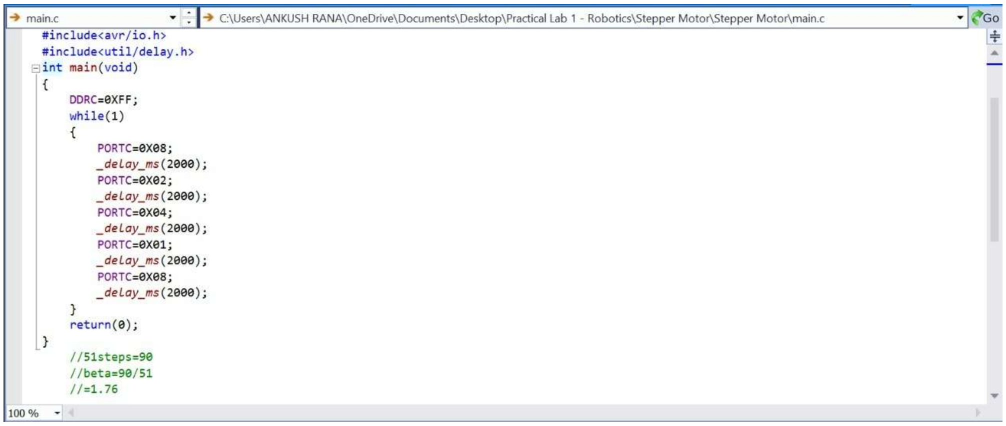
2. 2 phase full stepping motor

3. Half phase stepping motor



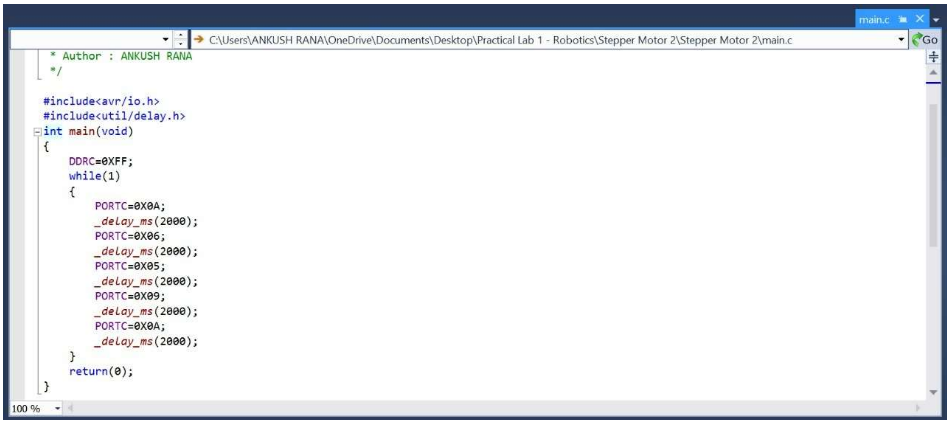
## SOURCE CODE: -

## 1.


main.c
C:\Users\ANKUSH RANA\OneDrive\Documents\Desktop\Practical Lab 1 - Robotics\Stepper Motor\Stepper Motor\main.c
Go
100 %

```
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
    DDRC=0XFF;
    while(1)
    {
        PORTC=0X08;
        _delay_ms(2000);
        PORTC=0X02;
        _delay_ms(2000);
        PORTC=0X04;
        _delay_ms(2000);
        PORTC=0X01;
        _delay_ms(2000);
        PORTC=0X08;
        _delay_ms(2000);
    }
    return(0);
}
    //51steps=90
    //beta=90/51
    //=1.76
```

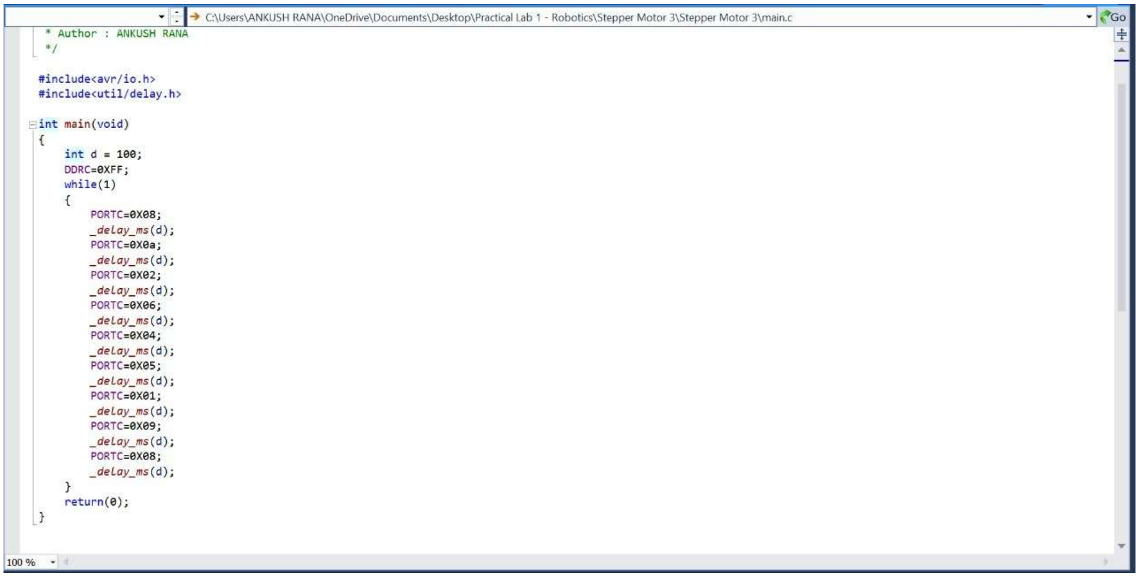
## 2.


main.c
C:\Users\ANKUSH RANA\OneDrive\Documents\Desktop\Practical Lab 1 - Robotics\Stepper Motor 2\Stepper Motor 2\main.c
Go
100 %

```
 * Author : ANKUSH RANA
 */

#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
    DDRC=0XFF;
    while(1)
    {
        PORTC=0X0A;
        _delay_ms(2000);
        PORTC=0X06;
        _delay_ms(2000);
        PORTC=0X05;
        _delay_ms(2000);
        PORTC=0X09;
        _delay_ms(2000);
        PORTC=0X0A;
        _delay_ms(2000);
    }
    return(0);
}
```

## 3.



C:\Users\ANKUSH RANA\OneDrive\Documents\Desktop\Practical Lab 1 - Robotics\Stepper Motor 3\Stepper Motor 3\main.c

```
 * Author : ANKUSH RANA
 */

#include<avr/io.h>
#include<util/delay.h>

int main(void)
{
    int d = 100;
    DDRC=0XFF;
    while(1)
    {
        PORTC=0X08;
        _delay_ms(d);
        PORTC=0X0a;
        _delay_ms(d);
        PORTC=0X02;
        _delay_ms(d);
        PORTC=0X06;
        _delay_ms(d);
        PORTC=0X04;
        _delay_ms(d);
        PORTC=0X05;
        _delay_ms(d);
        PORTC=0X01;
        _delay_ms(d);
        PORTC=0X09;
        _delay_ms(d);
        PORTC=0X08;
        _delay_ms(d);
    }
    return(0);
}
```

## CONCLUSION: -

A step per motor is been rotated by having different phase of operation.