LED Blinking with Push Button

Aim:- Write an embedded C program to interface LED and push button with Arduino UNO.

Software used:- Arduino IDE and Simulator

Program:-

```
// Program to interface LED with Arduino UNO
const int buttonPin = 4;     // Pin connected to pushbutton
const int ledPin = 12;      // Pin connected to LED
int buttonState; // Give pushbutton a value
void setup()
{
pinMode(ledPin, OUTPUT);  // Set LED pin as output
pinMode(buttonPin, INPUT); // Set pushbutton pin as input

 }

void loop() { buttonState = digitalRead(buttonPin); // Read input from pin 4
 if(buttonState == LOW) { // If pushbutton is pressed, set as low

 digitalWrite(ledPin, HIGH); // Turn on LED

delay((500));
 digitalWrite(ledPin, LOW);

 delay(500);
}
else
{
  digitalWrite(ledPin, LOW); // Otherwise, turn off LED

   }
   }
```

Code

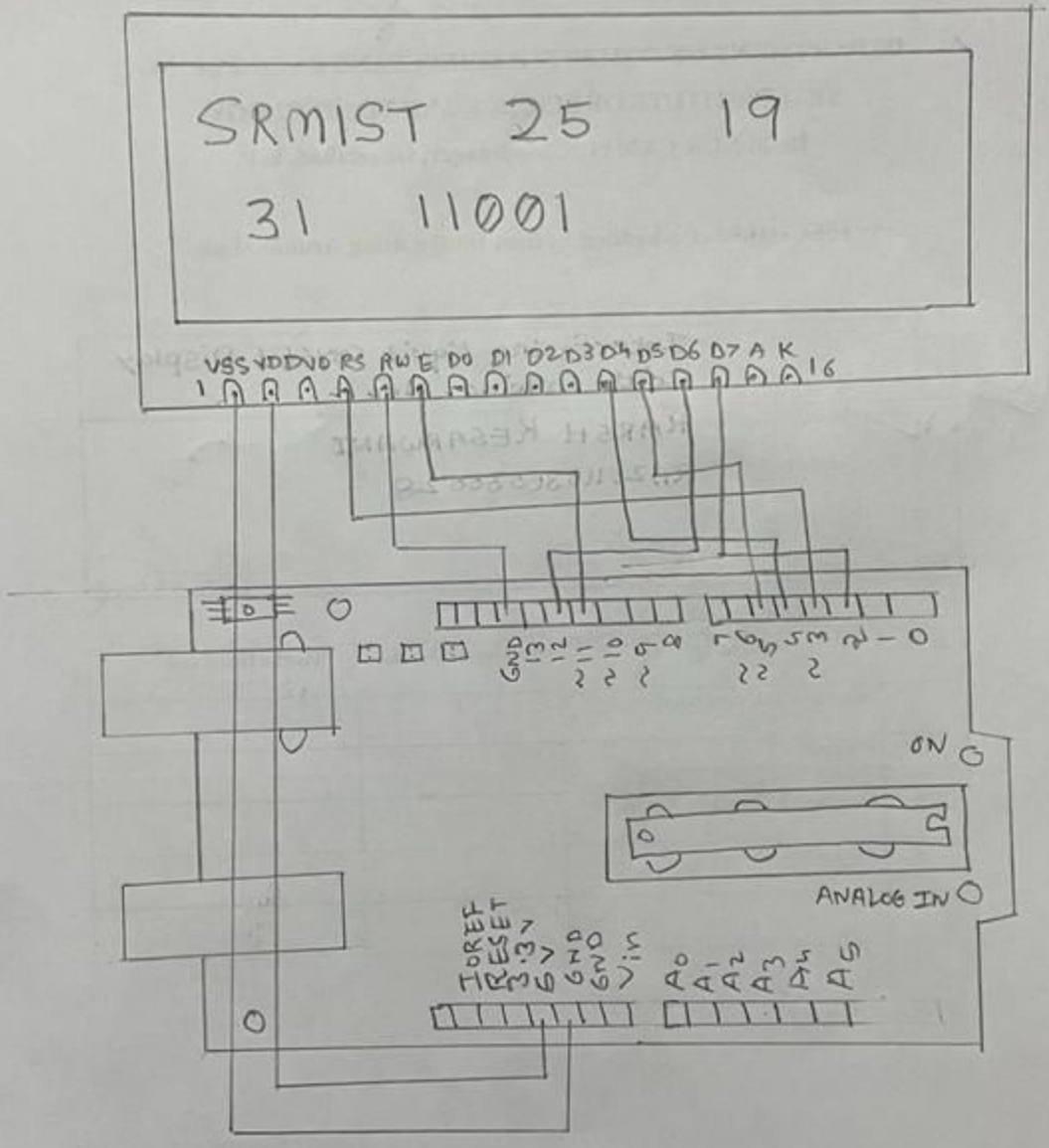
```
int a=13;
int b=12;
int c=11;
int d=10;
int e=9;
int f=8;
int g=7;

void setup() {
  // put your setup code here, to run once:
pinMode (a, OUTPUT);
pinMode (b, OUTPUT);
pinMode (c, OUTPUT);
pinMode (d, OUTPUT);
pinMode (e, OUTPUT);
pinMode (f, OUTPUT);
pinMode (g, OUTPUT);

}

void loop() {
  // put your main code here, to run repeatedly:

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,0);
digitalWrite (g,1);
delay (500);

digitalWrite (a,1);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,1);
digitalWrite (g,1);
delay (500);

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,1);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,1);
digitalWrite (g,0);
delay (500);

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,1);
digitalWrite (f,1);
digitalWrite (g,0);
delay (500);
```

```
delay (500);

digitalWrite (a,1);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,0);
digitalWrite (g,0);
delay (500);

digitalWrite (a,0);
digitalWrite (b,1);
```

```
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,1);
digitalWrite (f,0);
digitalWrite (g,0);
delay (500);

digitalWrite (a,0);
digitalWrite (b,1);
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,0);
digitalWrite (g,0);
delay (500);

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,1);
digitalWrite (g,1);
delay (500);

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,0);
```

```
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,1);
digitalWrite (f,0);
digitalWrite (g,0);
delay (500);

digitalWrite (a,0);
digitalWrite (b,1);
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,0);
digitalWrite (g,0);
delay (500);

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,1);
digitalWrite (g,1);
delay (500);

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,0);
digitalWrite (g,0);
delay (500);

digitalWrite (a,0);
digitalWrite (b,0);
digitalWrite (c,0);
digitalWrite (d,0);
digitalWrite (e,1);
digitalWrite (f,0);
digitalWrite (g,0);
delay (500);


}
```

Output



Interfacing Diagram

Experiment 3

AIM: Interfacing Liquid Crystal Display with Arduino Uno

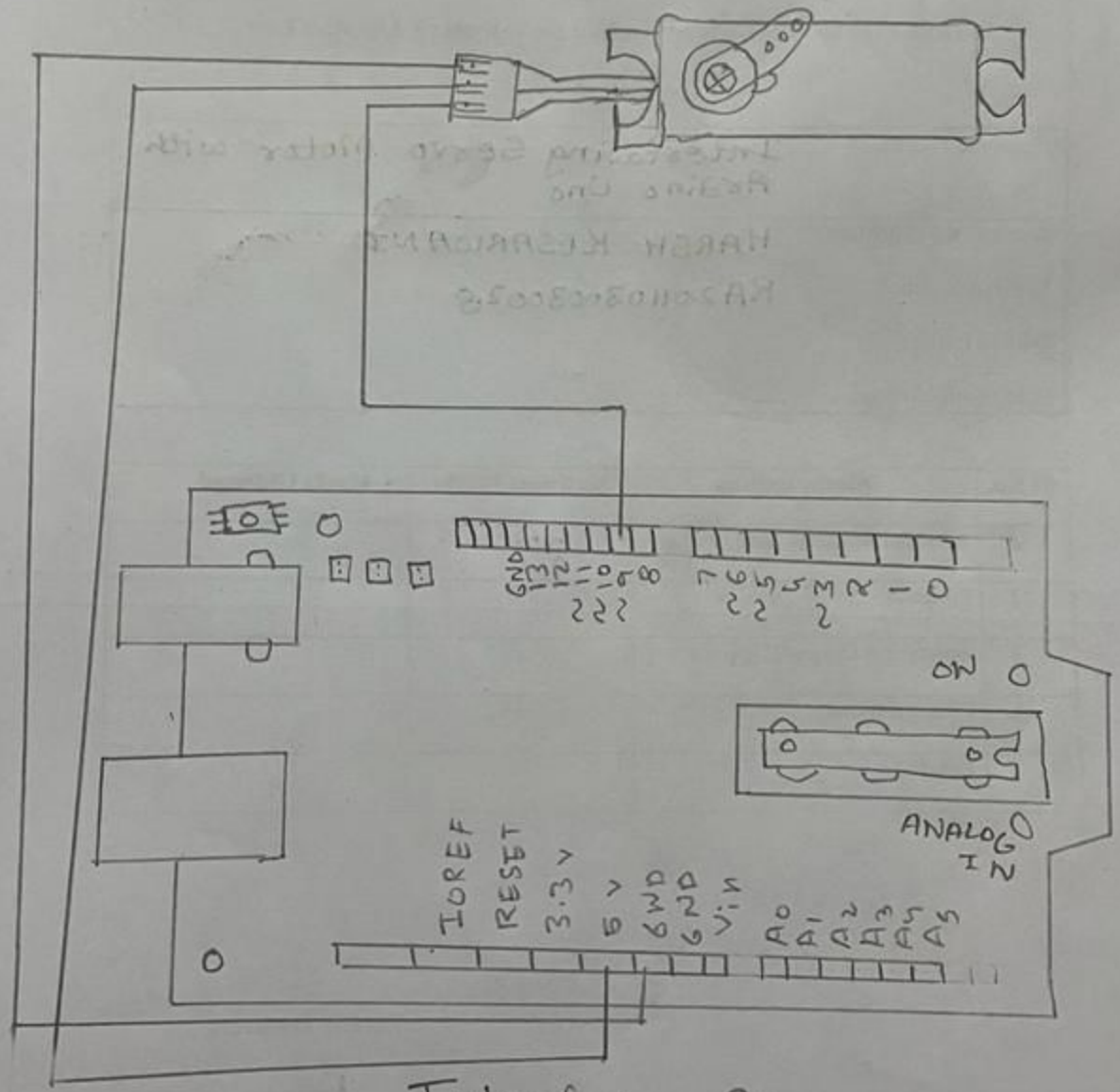
COMPONENTS USED: Arduino Uno, LCD (16x2) and wires

SOFTWARE USED: Wokwi

INTERFACING CODE

```
#include <LiquidCrystal.h>
int rs = 12, en=11, d4 = 5, d5=4, d6=3, d7=2;
LiquidCrystal lcd (rs,en,d4,d5,d6,d7);
void setup()
{
    lcd.begin(16,2);
}
void loop()
{
    lcd.home();
    lcd.println("SRMIST");
    delay(1000);
    lcd.println(25,DEC);
    delay(1000);
    lcd.println(25, HEX);
    delay(1000);
    lcd.setCursor(1,2);
    lcd.println(25,OCT);
    delay(1000);
    lcd.println(25, BIN);
    delay(1000);
}
```

RESULT: We have successfully interfaced the Liquid Crystal Display with Arduino Uno and it is showing the desired output.



Interfacing Diagram

Experiment 4

AIM: Interfacing Servo Motor with Arduino Uno

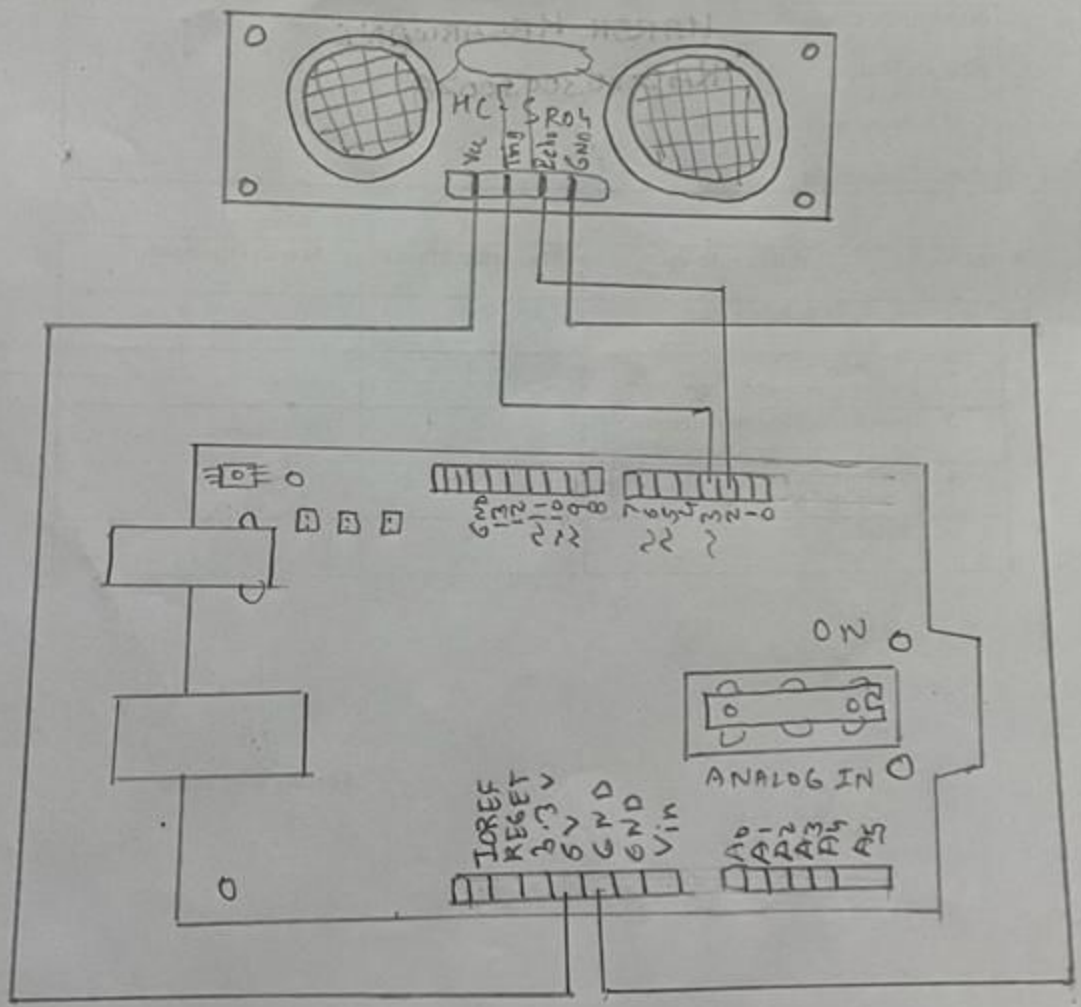
COMPONENTS USED: Arduino Uno, Servo Motor and Wires

SOFTWARE USED: Wokwi

INTERFACING CODE

```
#include <Servo.h>
Servo myservo; // create servo object to control a servo
// 12 servo objects can be created on most boards
int pos; // To store servo position
void setup()
{
    myservo.attach(9); // attaches servo on pin 9 to the
                       servo object
}
void loop()
{
   for (pos = 0; pos <= 180; pos += 1) // goes from 0 to 180 degrees
    {    //tell servo to go to position in variable pos
         myservo.write(pos);
         delay(15); // wait 15ms for the servo to reach
                      position
    }
    for (pos = 180; pos >= 0; pos -= 1) // goes from 180 to
                                           0 degrees
    {
         myservo.write(pos);
         delay(15);
    }
}
```

RESULT: We have successfully interfaced servo motor with Arduino Uno and it is showing the desired movement from 0 to 180 degrees and from 180 to 0 degrees.



Interfacing Diagram

Experiment 5

Aim: Interfacing Ultrasonic Sensor with Arduino Uno

Components Used: Arduino Uno, Ultrasonic Sensor and wires

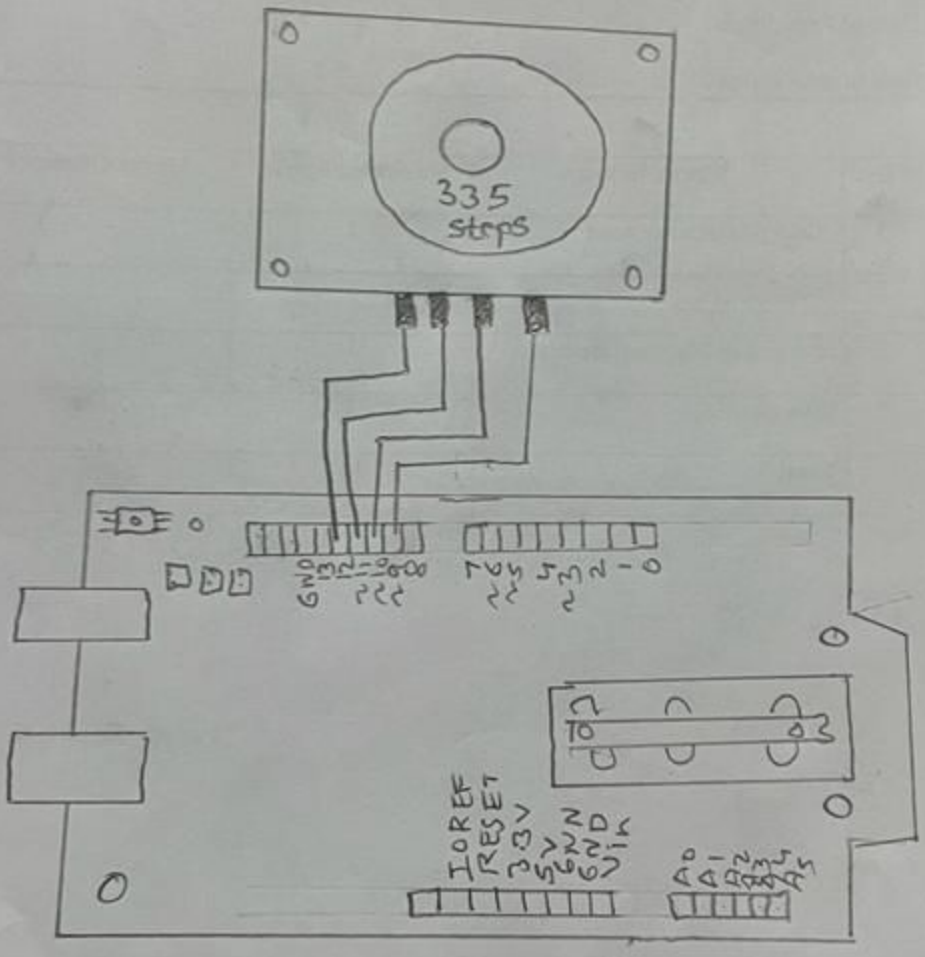
Software Used: Wokwi

Interfacing Code

```
#define echoPin 2
#define trigPin 3

float duration;
float distance;
void setup()
{
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}
void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = (duration * 0.034312);
    Serial.print("Distance : ");
    Serial.print(distance);
    Serial.println(" cm");
```

```
    delay(1000);
}
```

RESULT: We have successfully interfaced ultrasonic sensor with arduino Uno and it is giving the desired outcome.



clockwise
Counterclockwise

Interfacing
Diagram

Experiment 6

AIM: Interfacing Bipolar Stepper Motor with Arduino Uno

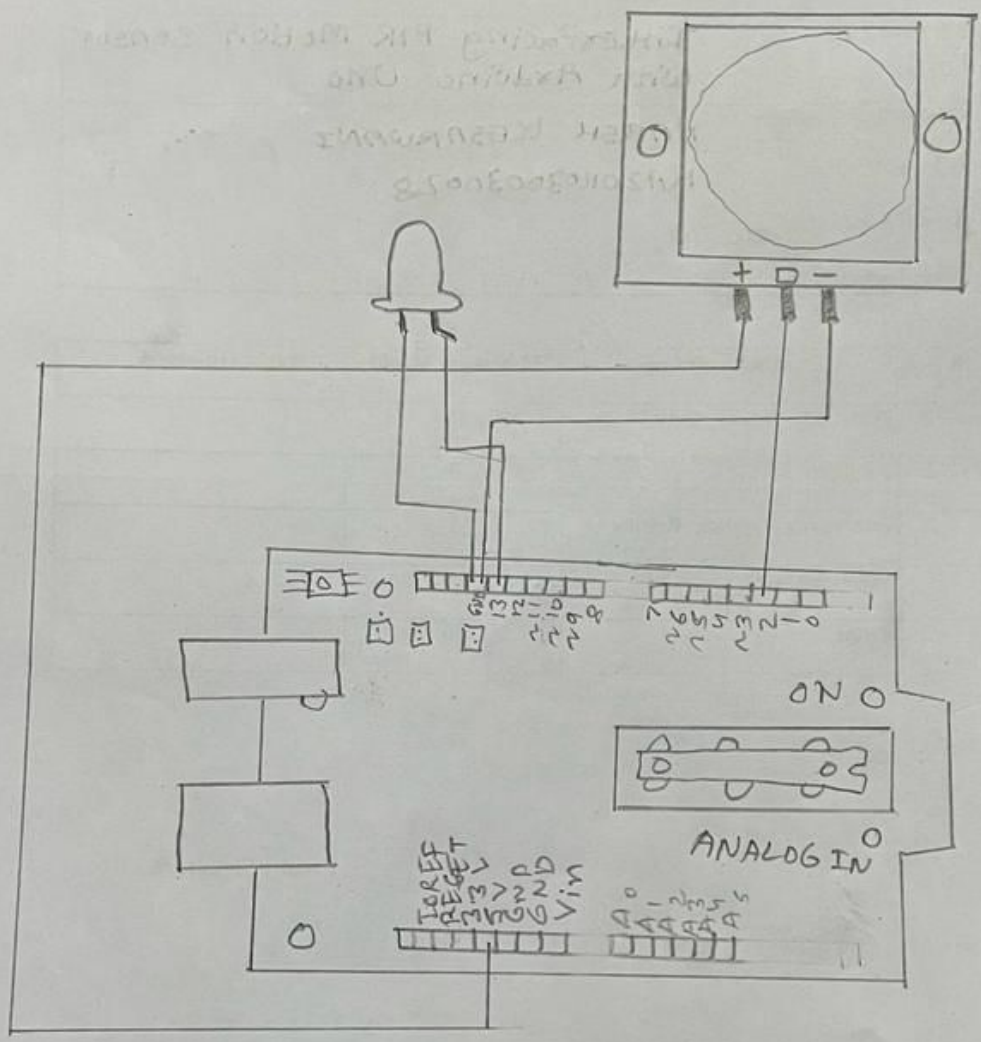
COMPONENTS USED: Arduino Uno, Bipolar Stepper Motor and wires.

SOFTWARE USED: Wokwi

INTERFACING CODE:

```
#include <Stepper.h>
Const int StepsPerRevolution = 200; // change this to fit the number of steps per revolution for your motor
Stepper myStepper (stepsPerRevolution, 8, 9, 10, 11);
void setup() //initialize the stepper library on pins 8 through 11
{
    myStepper.setSpeed(60); // set the speed at 60 rpm
    Serial.begin(9600); //initialize serial port
}
void loop()
{
    // step one revolution in one direction
    Serial.println("clockwise");
    myStepper.step(400);
    delay(500);

    // step one revolution in another direction
    Serial.println("counterclockwise");
    myStepper.step(-400);
    delay(500);
}
```

RESULT: We have successfully interfaced bipolar stepper motor with Arduino Uno and it is showing the desired output.



Motion detected!
Motion ended!

Interfacing Diagram

Experiment 7

Aim: Interfacing PIR Motion Sensor with Arduino Uno

Components Used: Arduino Uno, PIR Motion Sensor and wires

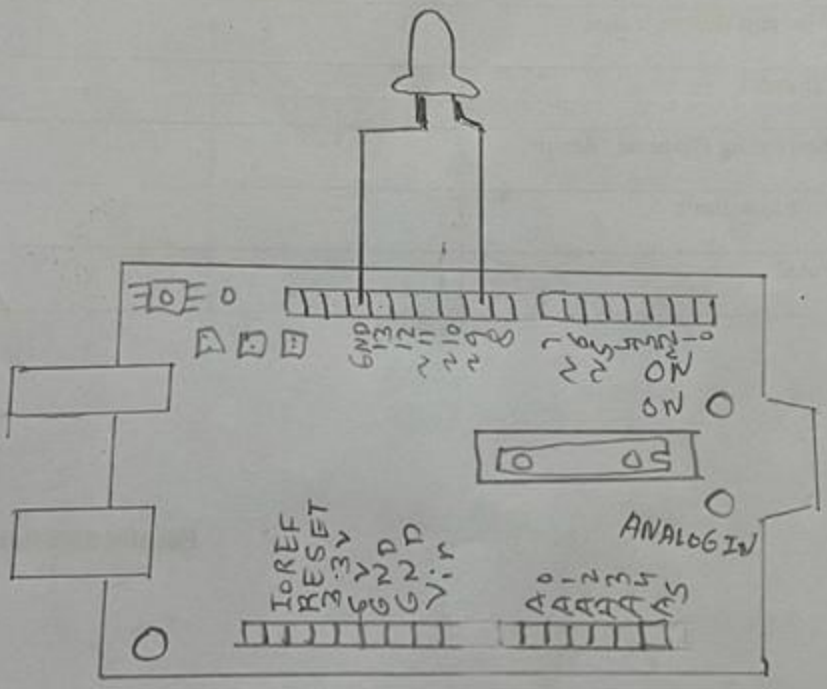
Software Used: Wokwi

Interfacing Code:

```
int ledPin = 13;  // Pin for LED
int inputPin = 2; // input Pin for PIR
int pinState = LOW; // we start assuming, no motion detected
int val = 0;   // for reading pin status
void setup()
{
    pinMode(ledPin, OUTPUT); // declared LED as output
    pinMode(ledPin, INPUT); // declared sensor as input
    Serial.begin(9600);
}
void loop()
{
    val = digitalRead(inputPin); // read input value
    if (val == HIGH) // check if input is high
    {
        digitalWrite(ledPin, HIGH); // turn LED on
        if(pinState == LOW)
        {   // we have just turned on
            Serial.println("Motion detected!");
            // we only wont to print on the output change
            pinState = HIGH;
        }
    }
    else
    {
```

```
        digitalWrite (ledPin, LOW); // turn LED OFF
        if (pinState == HIGH)
        {   //we have just turned off
            Serial.println ("Motion ended!");
// we only want to print output change, not state
            pinState = LOW;
        }
    }
}
```

RESULT: We have successfully interfaced PIR Motion Sensor with Arduino Uno and it is giving desired output.



Interfacing Diagram

Experiment 8

AIM: PWM or Pulse Width Modulation using Arduino Uno

COMPONENTS USED: Arduino Uno, LED and wires

SOFTWARE USED: Wokwi

INTERFACING CODE:

```
void setup()
{

}
void loop()
{
    analogWrite(9,64);
    delay(1000);
    analogWrite(9,127);
    delay(1000);
    analogWrite(9,191);
    delay(1000);
    analogWrite(9,255);
    delay(1000);
    analogWrite(9,0);
    delay(1000);
}
```

RESULT: We have successfully interfaced LED with Arduino Uno to show PWM and it is giving desired result.