IOT PRACTICAL FILE

Bachelor of Computer Applications (BCA)

To

Guru Gobind Singh Indraprastha University

Submitted to:

Mr. Akshit Thakur

(Assistant Prof.)

Submitted By:

Aditya Kumar

Roll No. 04611102021

Banarsidas Chandiwala Institute of Information Technology

New Delhi – 110019

Batch (2021-2024)

BCA (307)

S.No	Practical List	Page No.
1	Introduction to basic electronic components used in IoT using a Simulator (Tinkercad)	
2	Introduction to Arduino its pin diagram and its components	
3	Introduction to Tinkercad and make basic (non-programmable) connections onTinkercad circuit simulation.	
4	Study and Install IDE of Arduino	
5	Write the steps to add libraries in Arduino and setup of Arduino IDE forprogramming.	
6	Write a Program using Arduino for Blink LED's	
7	Write a Program to show how to fade an LED on PWM pin of Arduino using analogWrite() function and also attach oscilloscope for graphical representation.	
8	Control directions (clockwise/anticlock) and speed of hobby gearmotor using L293D IC with Arduino board.	
9	Design a circuit to vary the light intensity of LED using Photoresistor sensor with Arduino.	
10	Integrating temperature sensor throught arduino boardand its application.	
11	Integrating gas sensor in arduino board designing asmoke detector.	
12	Integrating PIR sensor with arduino usin tinkercad simulator and print its value on serial monitor .	
13	Measuring the distance using ultrasonic sensors and make LEDs blink using arduino. (Car Parking system)	
14	Home automation system (Sensors: Ultrasonic, PIR, Temperature sensor Actuators: Servomotor, DC-motor, LED's)	

Q1. Introduction to basic electronic components used in IoT using a Simulator (Tinkercad)

ANS:

Arduino Uno R3:

The Arduino Uno R3 is a widely-used microcontroller board known for its versatility and ease of use. It features the ATmega328P microcontroller chip, 14 digital input/output pins, 6 analog input pins, USB connectivity for programming and serial communication, and compatibility with a wide range of shields and sensors. With its open-source design and beginner-friendly programming environment, the Uno R3 is a popular choice for hobbyist projects, educational purposes, and prototyping applications.

Ultrasonic Distance Sensor:

Ultrasonic distance sensors use sound waves to measure distance. They emit ultrasonic pulses and measure the time it takes for the pulses to bounce back after hitting an object. This data is then used to calculate the distance to the object. These sensors are commonly used in robotics, automation, and security systems for obstacle detection, distance measurement, and object tracking.

Photoresistor (Light Dependent Resistor):

A photoresistor, or LDR, is a type of resistor whose resistance decreases with increasing light intensity. This means it can be used to detect light levels. When exposed to light, the resistance of an LDR decreases, allowing more current to flow through it. Photoresistors find applications in light-sensitive circuits, such as streetlights, camera light meters, and automatic night lights.

Gas Sensor:

Gas sensors are used to detect the presence of gases in the surrounding environment. They work by reacting to specific gases and producing an electrical signal proportional to the concentration of the target gas. Different types of gas sensors are available, each designed to detect specific gases such as carbon monoxide, methane, or carbon dioxide. These sensors are commonly used in industrial safety systems, environmental monitoring, and indoor air quality monitoring.

PIR Sensor (Passive Infrared Sensor):

PIR sensors detect motion by measuring changes in infrared radiation emitted by objects in their field of view. They consist of a pyroelectric sensor that generates an electrical signal when exposed to infrared radiation emitted by warm objects, such as humans or animals. PIR sensors are commonly used in security systems, automatic lighting systems, and occupancy detection systems to detect motion and trigger appropriate actions, such as turning on lights or sounding alarms.

LED (Light Emitting Diode):

LEDs are semiconductor devices that emit light when an electric current passes through them. They are widely used as visual indicators in electronic devices, status indicators in circuits, and for general illumination in various applications. LEDs are energy-efficient, long-lasting, and available in a variety of colors and sizes, making them versatile for use in lighting, displays, and decorative purposes.

DC Motor:

DC motors convert electrical energy into mechanical energy through the interaction of magnetic fields. They consist of a stationary part called the stator and a rotating part called the rotor. When a current is applied to the motor, it creates a magnetic field that interacts with the magnetic field of the rotor, causing it to rotate. DC motors are used in a wide range of applications, including robotics, industrial automation, consumer electronics, and automotive systems.

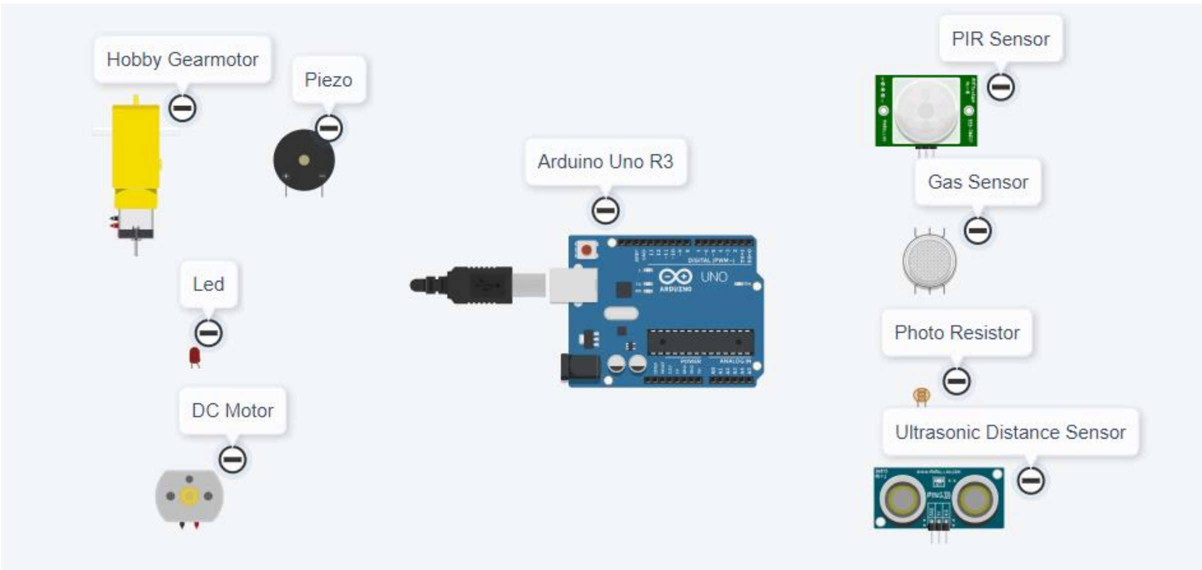
Hobby Gearmotor:

Hobby gearmotors are compact DC motors equipped with gearboxes to increase torque and reduce speed. These gearmotors are commonly used in hobbyist projects, robotics, and model building, where precise control of speed and torque is required. The gearbox attached to the motor helps to amplify the motor's torque while reducing its speed, making it suitable for driving wheels, gears, and other mechanical components in small-scale projects.

Piezoelectric Transducer:

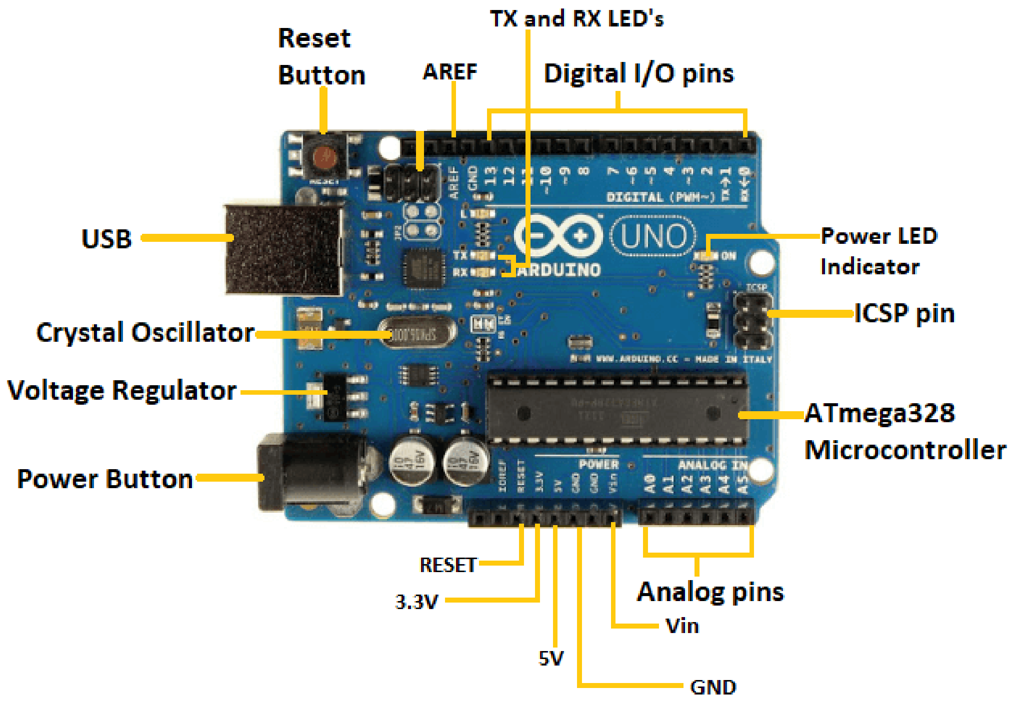
Piezoelectric transducers are devices that convert electrical energy into mechanical vibrations or vice versa. They utilize the piezoelectric effect, where certain materials produce an electric charge when subjected to mechanical stress. Piezoelectric transducers are used in various applications, including buzzers, speakers, microphones, sensors, and actuators. They are valued for their compact size, fast response time, and high-frequency capabilities, making them suitable for applications requiring precise control of mechanical motion or acoustic signals.

OUTPUT:



Q2. Introduction to Arduino its pin diagram and its components

Ans:



Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's designed for artists, designers, hobbyists, and anyone interested in creating interactive projects. Arduino boards are able to read inputs—light on a sensor, a finger on a button, or a Twitter message—and turn it into an output—activating a motor, turning on an LED, publishing something online. All this is possible with the help of a simple programming language and development environment that anyone can learn.

Components of an Arduino Board:

Microcontroller: At the heart of an Arduino board is a microcontroller, which acts as its brain. The most commonly used microcontroller in Arduino boards is from the Atmel AVR family, like the ATmega328P, ATmega2560, etc.

Power Supply: Arduino boards can be powered via USB connection or an external power source like batteries. They typically accept a wide range of voltages.

Input/Output (I/O) Pins: These are the digital and analog pins on the Arduino board. They allow the board to interact with the outside world. Digital pins can be programmed to read digital signals (HIGH or LOW), while analog pins can read analog voltage levels.

Voltage Regulator: Arduino boards usually have a voltage regulator that ensures a stable voltage supply to the microcontroller and other components.

Clock Crystal: This component provides the clock signal needed for the microcontroller to execute instructions accurately.

Reset Button: A button that allows you to reset the microcontroller, restarting your code execution from the beginning.

Q3.Introduction to Tinkercad and make basic (nonprogrammable) connections onTinkercad circuit simulation.

Ans:

Tinkercad is a user-friendly, web-based application for 3D modeling and design. It's particularly popular among beginners, educators, and hobbyists because of its simplicity and intuitive interface. Here's a brief introduction to Tinkercad:

Getting Started: To begin using Tinkercad, you just need to sign up for an account on their website. Once logged in, you'll be taken to the main workspace where you can start creating.

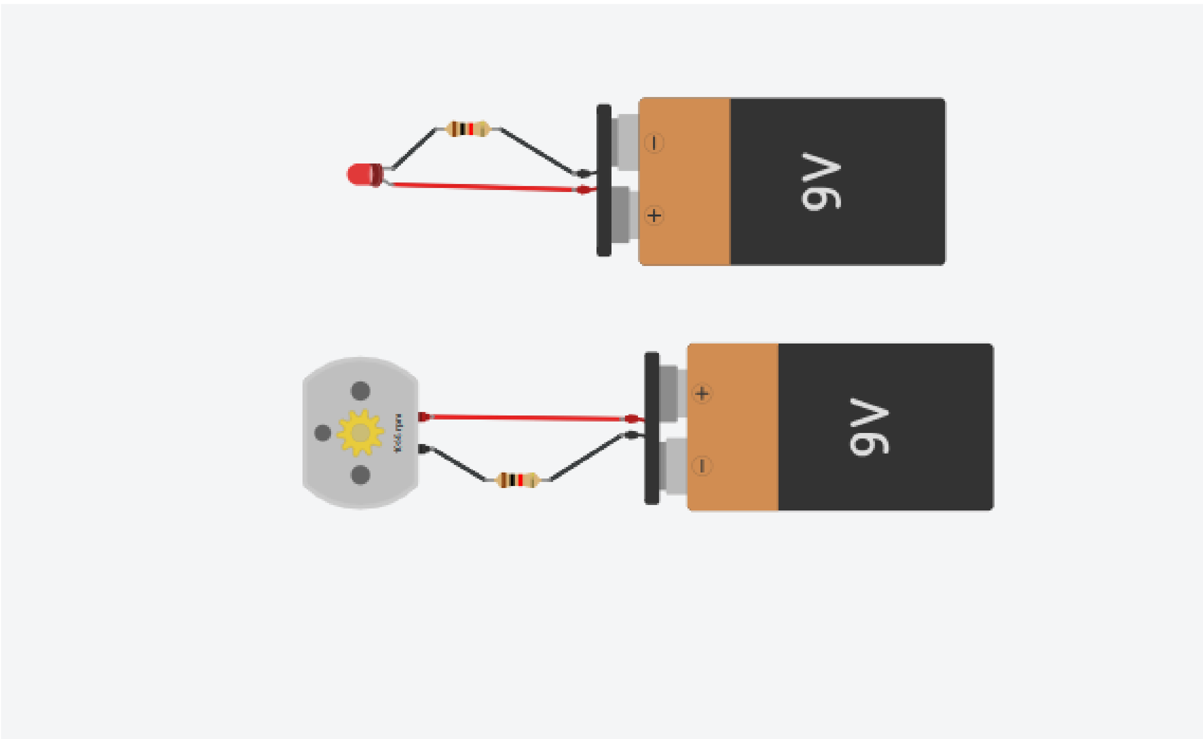
Creating Designs: Tinkercad allows you to create 3D designs using simple shapes like cubes, cylinders, and spheres. You can drag and drop these shapes onto the workspace and manipulate them in various ways to build your desired design.

Manipulating Objects: Tinkercad provides easy-to-use tools for resizing, rotating, and positioning objects. You can also group and ungroup objects to manipulate them as a single unit or individually.

Adding Details: Beyond basic shapes, Tinkercad offers features like text, holes, and custom shapes to add intricate details to your designs.

Color and Texture: You can apply different colors and textures to your objects to enhance their appearance. This can be done easily using Tinkercad's color picker and material options.

OUTPUT:



Q4. Study and Install IDE of Arduino

Ans:-

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

to create a new project, select File → **New**.

To open an existing project example, select File → Example → Basics → Blink.

Q5. Write the steps to add libraries in Arduino and setup of Arduino IDE for programming.

Ans:-

Step1- Open the Library Manager:

In the menu bar, select Tools > Manage Libraries…

In IDE 2, you can also click on the Library Manager icon button in the sidebar.

Step 2- Filter the available libraries by typing something (such as a library name) in the text field above the listed libraries.

Step 3- Find the library in the search results. The results are listed alphabetically, so you may need to scroll down the list.

Step 4- Find a library you want to install. You can review the description and author. When you've found a library you want to install, click Install. The latest version is selected by default.

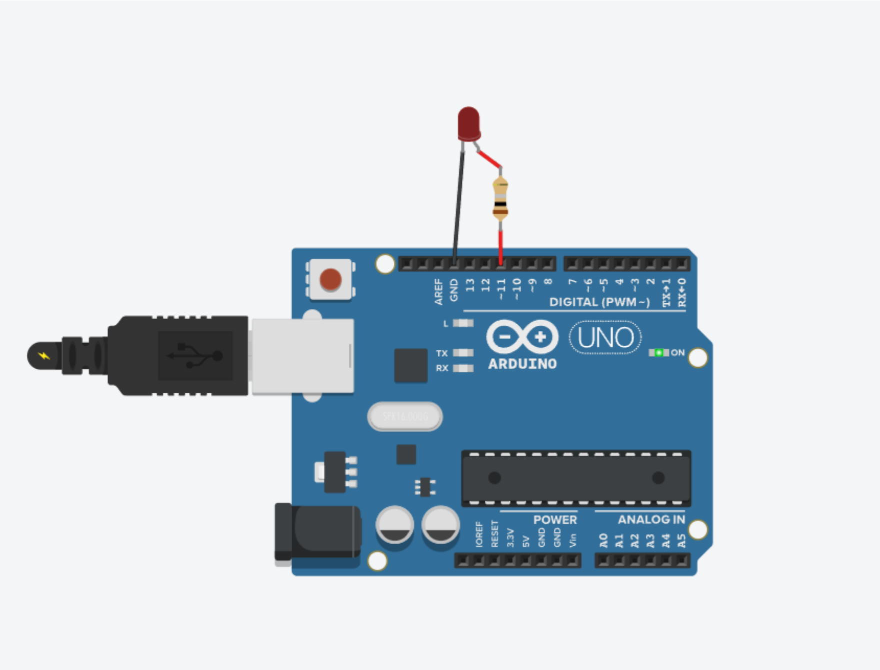
Step 5- Wait for the installation to complete.

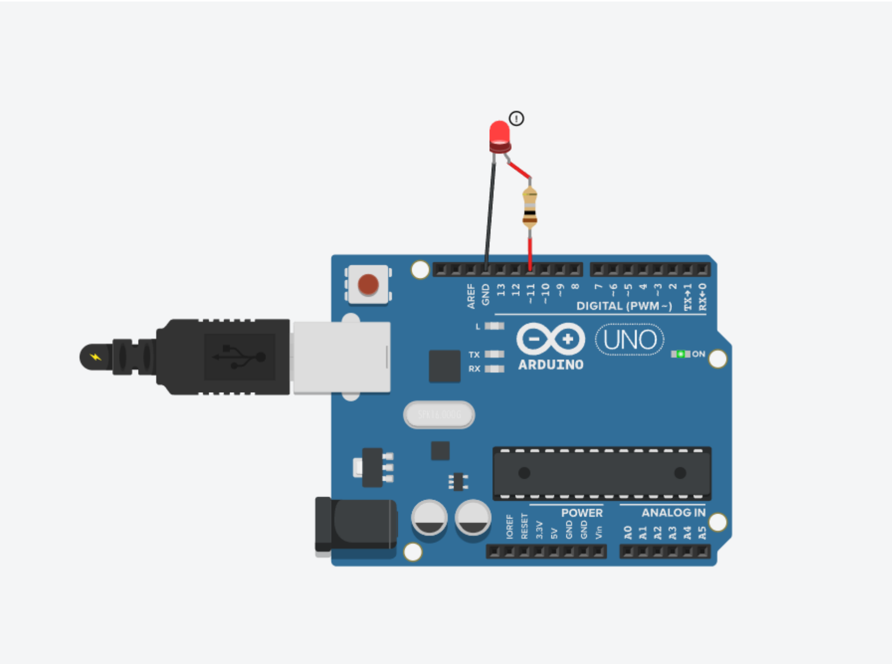
Q6. Write a Program using Arduino for Blink LED.

Ans:- //

```
int counter;
void setup()
{
pinMode(11, OUTPUT);
  for (counter = 0; counter < 10; ++counter) {
digitalWrite(11, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(11, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(11, HIGH);
  }
}
void loop()
{
  delay(10); // Delay a little bit to improve simulation performance
}
```

OUTPUT:


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM ~)
L
TX
RX
UNO
ARDUINO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

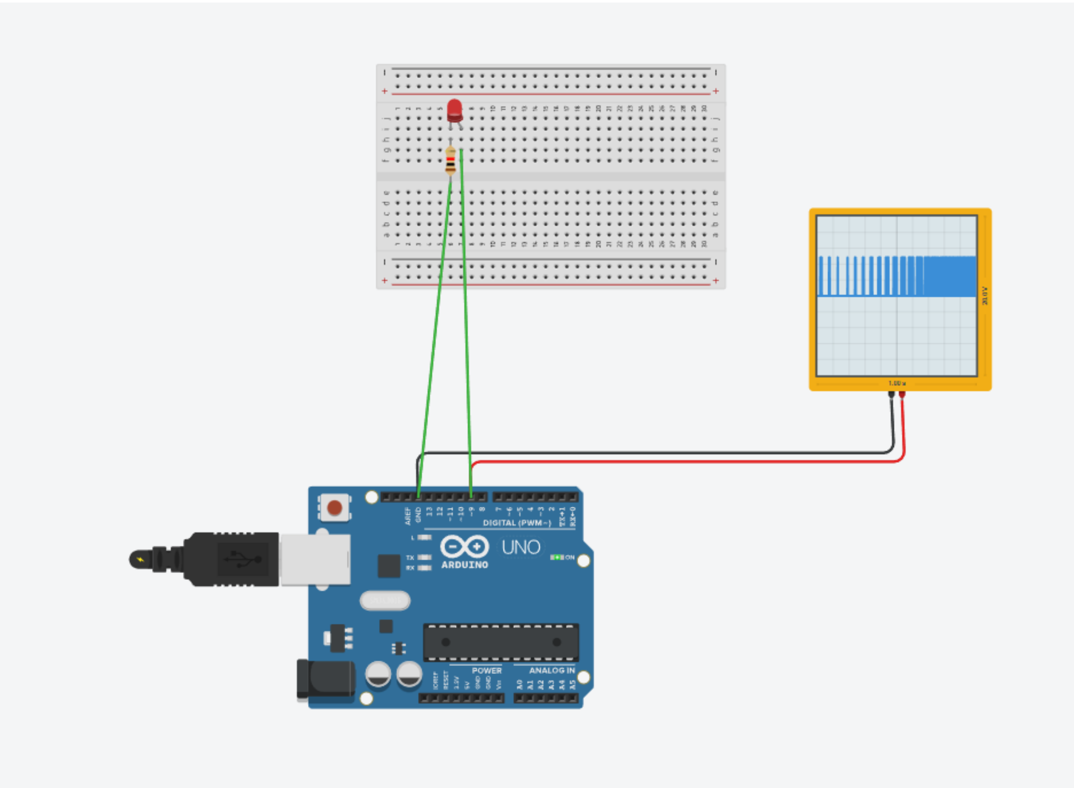

AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM ~)
L
TX
RX
UNO
ARDUINO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

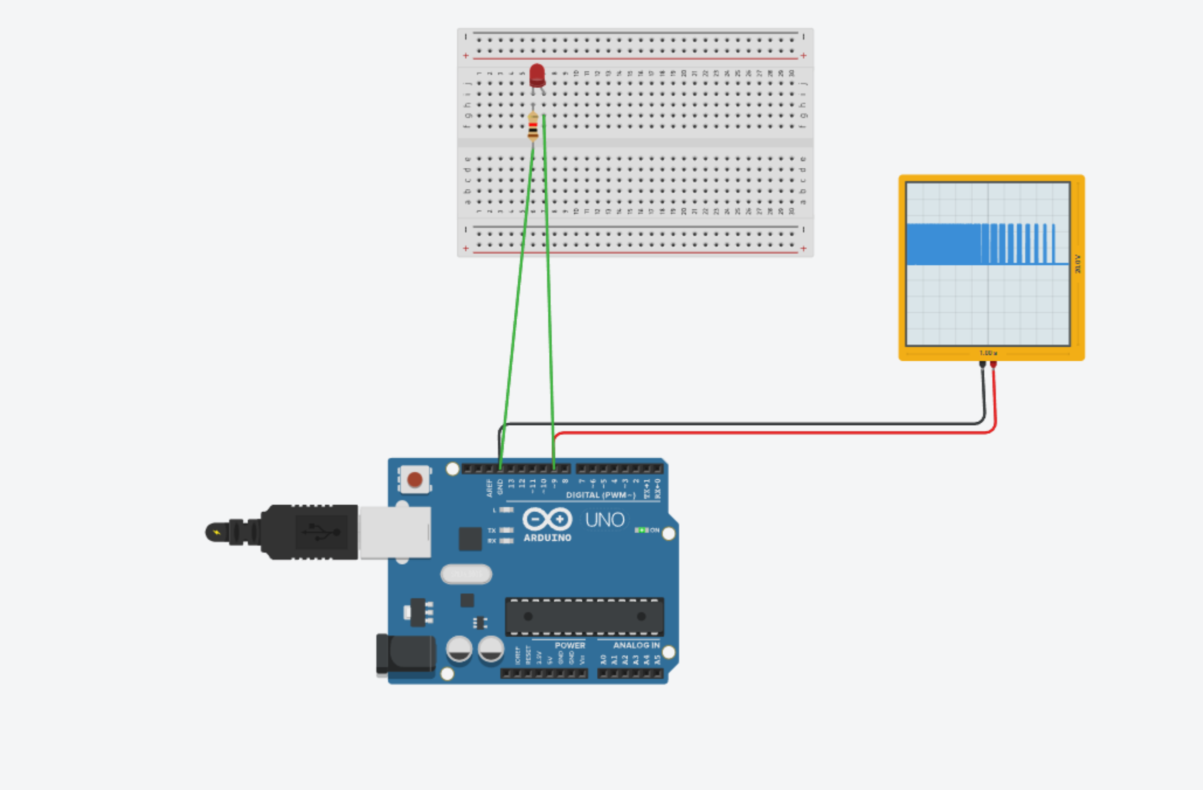
Q7.Write a Program to show how to fade an LED on PWM pin of Arduino using analogWrite() function and also attach oscilloscope for graphical representation.

CODE:

```
int led = 9;
int brightness = 0;
int fadeAmount = 5;
void setup() {
pinMode(led, OUTPUT);
}
void loop() {
analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness <= 0 || brightness >= 255) {
fadeAmount = -fadeAmount;
  }
  delay(30);
}
```

OUTPUT:



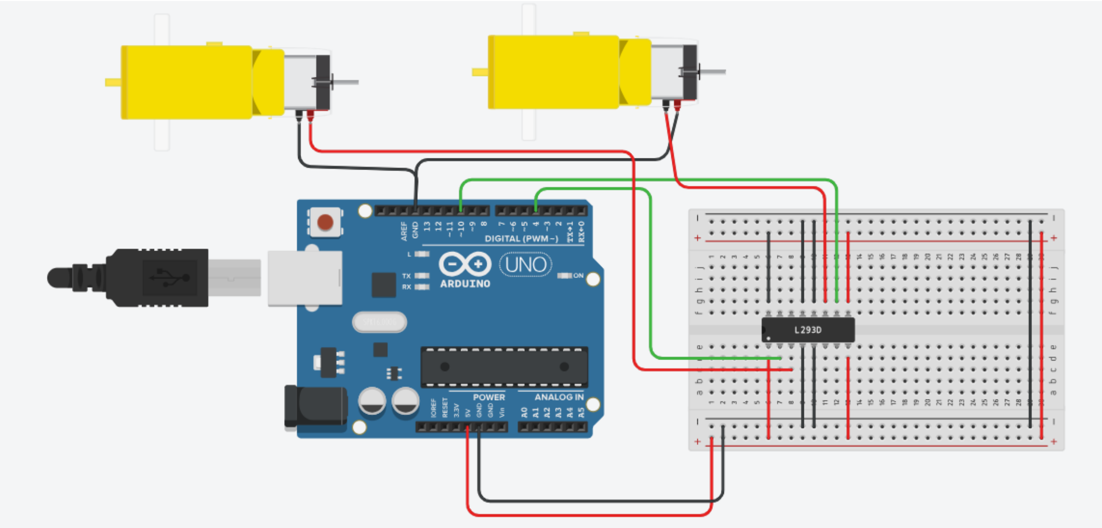


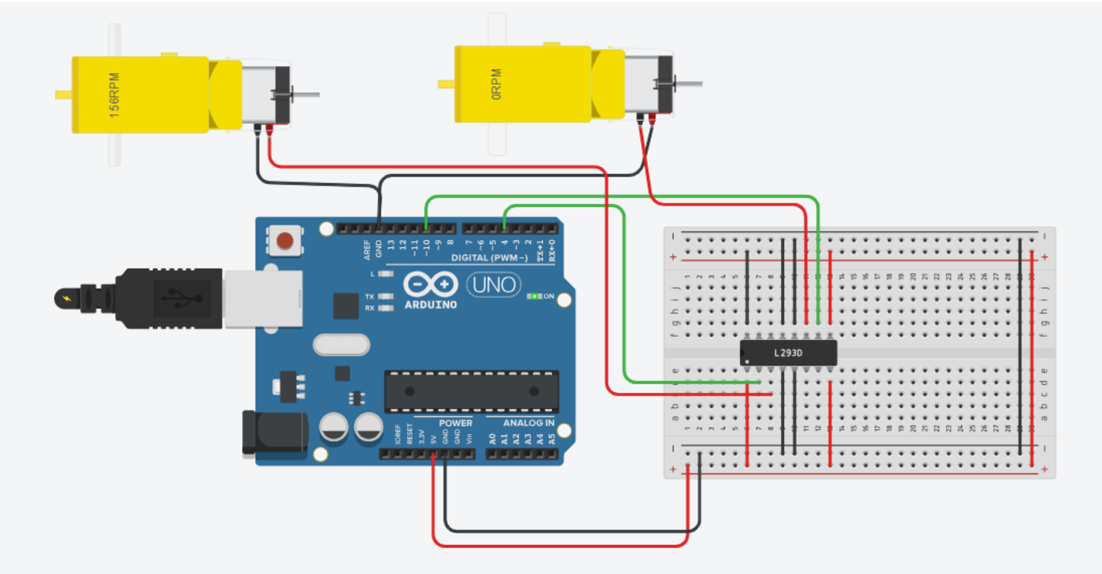
Q8. Control directions (clockwise/anticlock) and speed of hobbygearmotor using L293D IC with Arduino board.

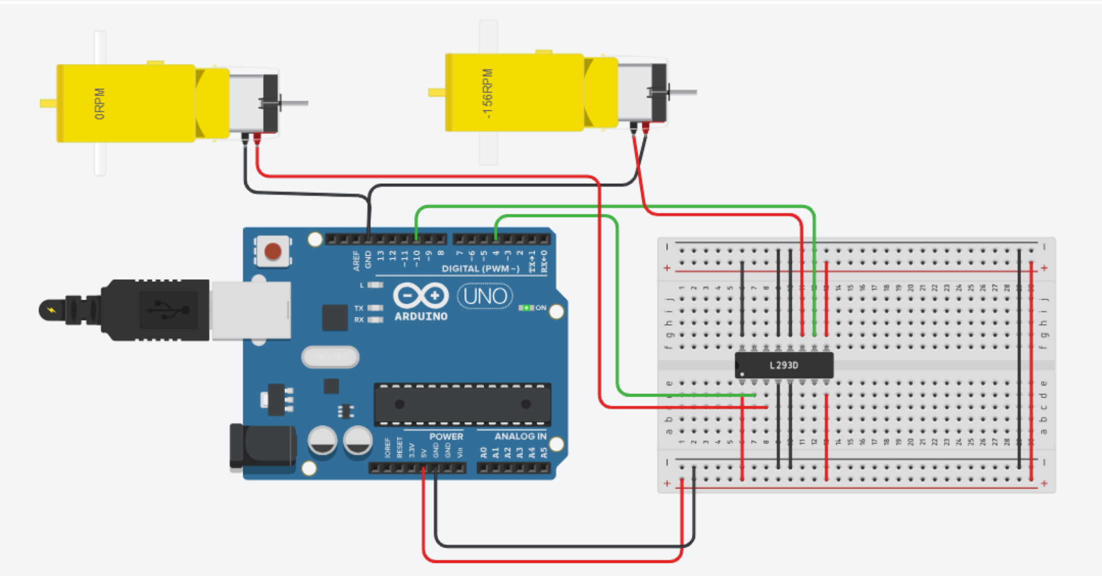
CODE:

```
void setup()
{
  pinMode(4, OUTPUT);
  pinMode(10, OUTPUT);
}
void loop()
{
  digitalWrite(4, HIGH);
  digitalWrite(10, LOW);
  delay(3000); // Wait for 3000 millisecond(s)
  digitalWrite(4, LOW);
  digitalWrite(10, HIGH);
  delay(3000); // Wait for 3000 millisecond(s)
}
```

OUTPUT:


UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
L293D


156RPM
0RPM
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
L293D


0RPM
-156RPM
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
L293D

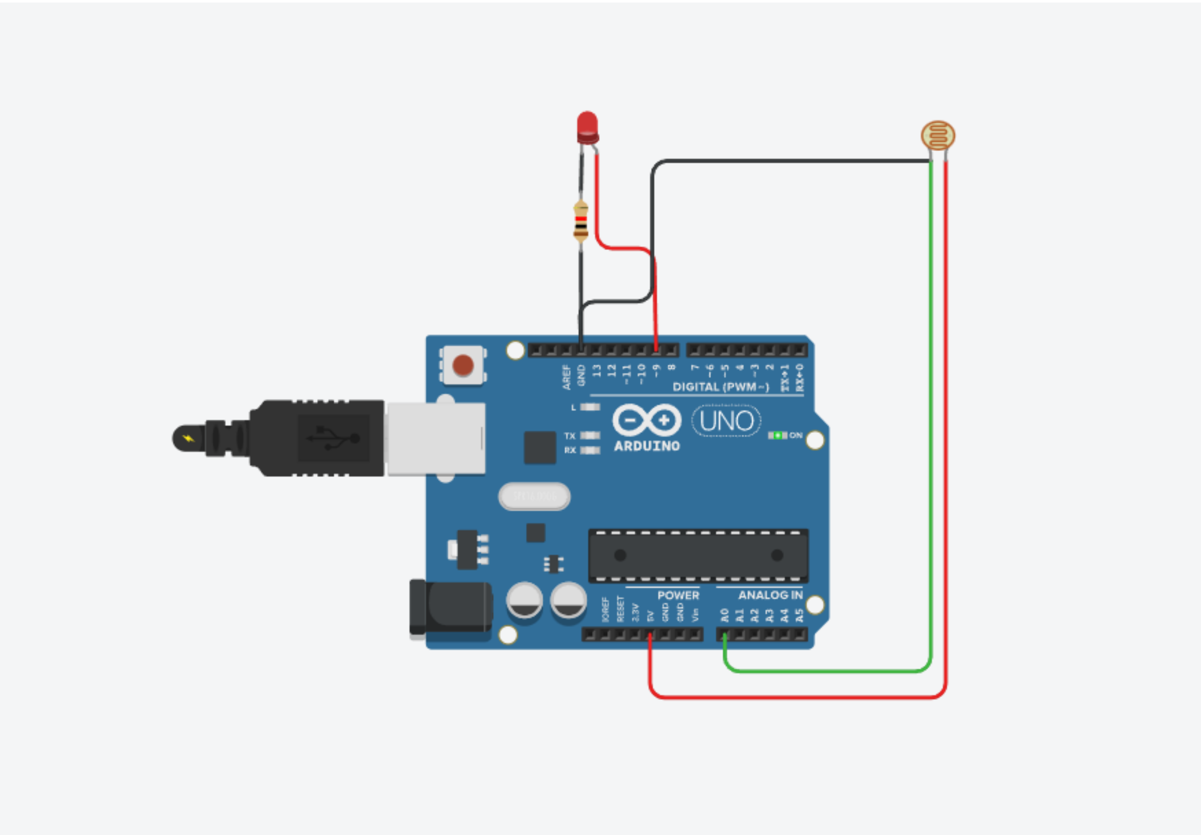
Q9.Design a circuit to vary the light intensity of LED using

Photoresistor sensor with Arduino.

CODE:

```
const int ledPin = 9;
const int  ldrPin = A0;
void setup() {
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
pinMode(ldrPin,  INPUT);
}
void loop() {
  int  ldrStatus = analogRead(ldrPin);
Serial.println(ldrStatus);
   if (ldrStatus<=80) {
digitalWrite(ledPin,  HIGH);
   }
  else {
digitalWrite(ledPin,  LOW);
  }
}
```

OUTPUT:

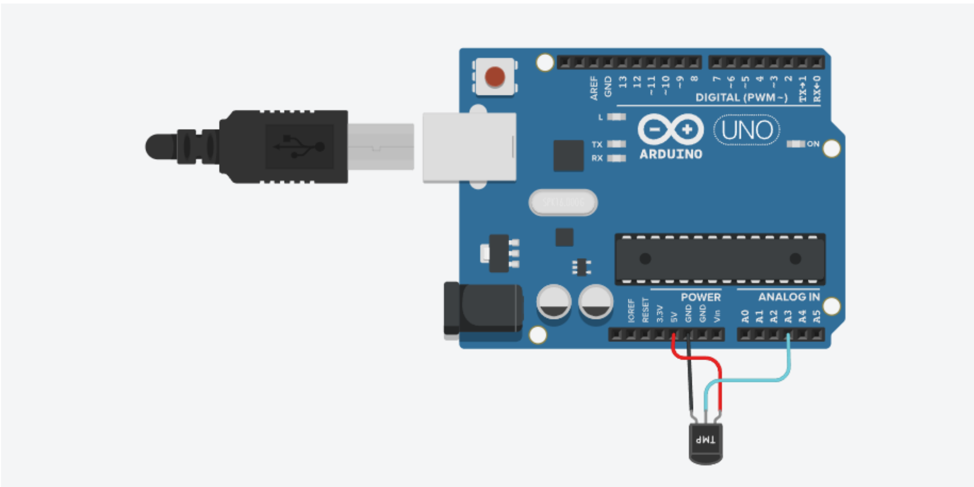


Q10. Write a Program for monitoring Temperature using Arduino and LM35 Temperature Sensors.

CODE:-

```
float temp;
int tempPin = 0;
void setup() {
Serial.begin(9600);
}
void loop() {
  temp = analogRead(tempPin);
  // read analog volt from sensor and save to variable temp
  temp = temp * 0.48828125;
  // convert the analog volt to its temperature equivalent
Serial.print("TEMPERATURE = ");
Serial.print(temp); // display temperature value
Serial.print("*C");
Serial.println();
  delay(1000); // update sensor reading each one second
}
```

Output:



Serial Monitor

```
TEMPERATURE = 186.04*C
TEMPERATURE = 123.05*C
TEMPERATURE = 351.56*C
TEMPERATURE = 96.19*C
TEMPERATURE = 437.01*C
```

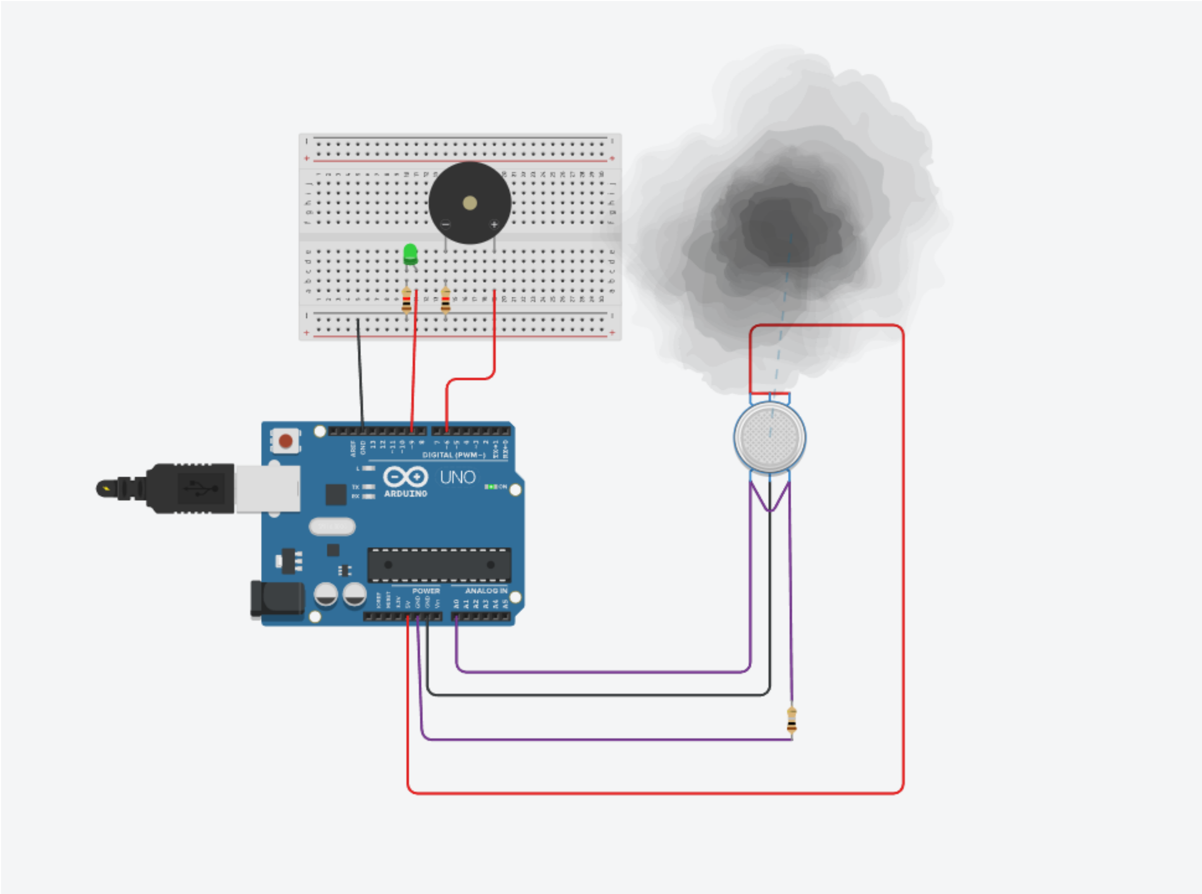
Q11.Integrating gas sensor in arduino board designing asmoke

Detector

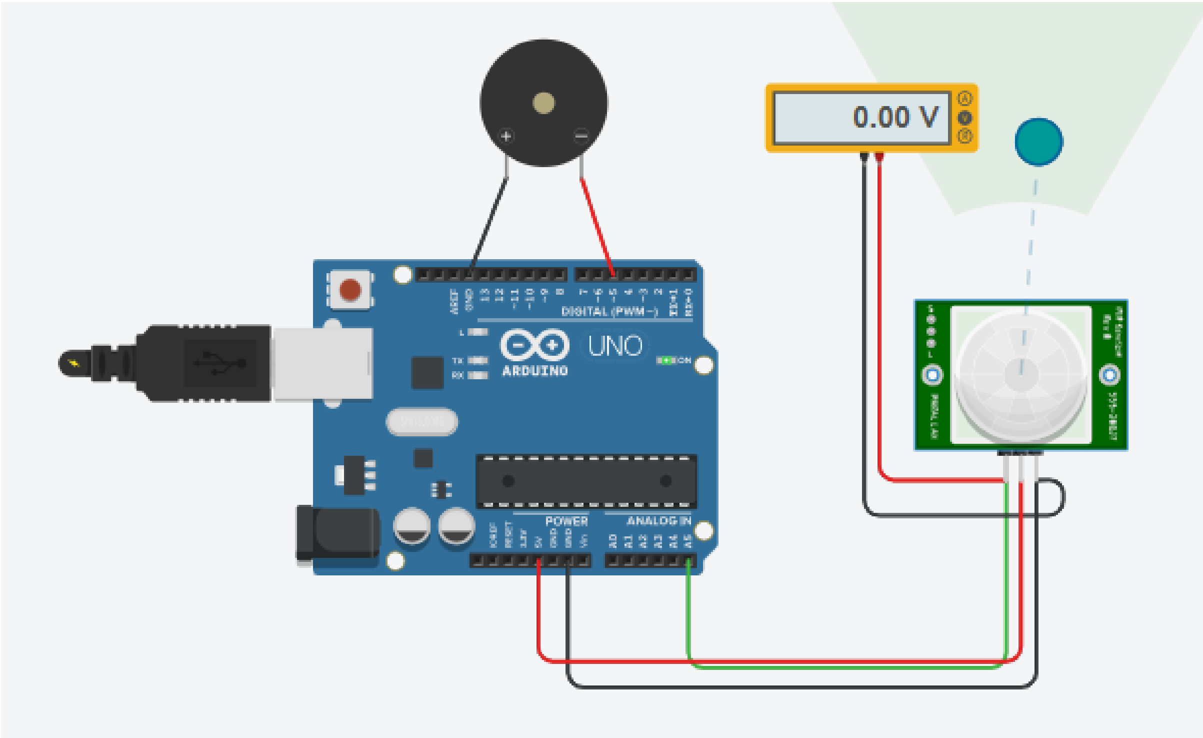
CODE:

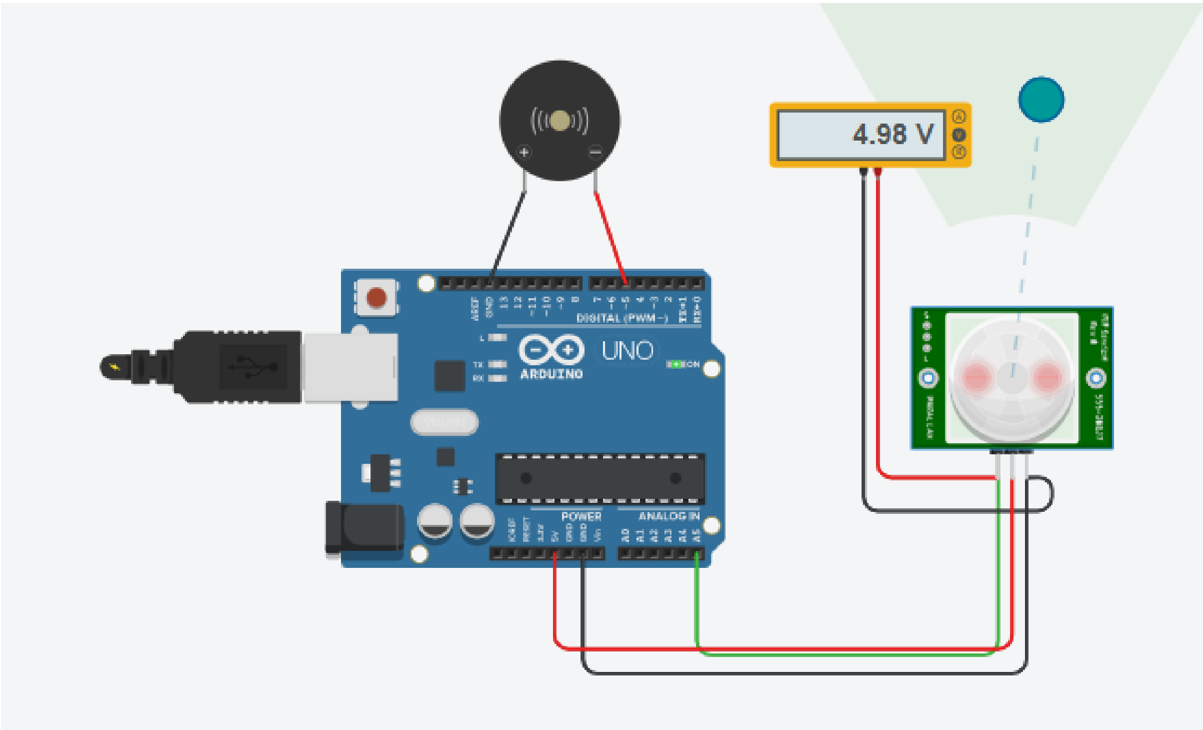
```
int red_LED_PIN = 11;
int green_LED_PIN = 9;
int blue_LED_PIN =  10;
int buzzer = 6;
int smoke_detector = A0;
int safety_lim = 60;
void setup() {
pinMode(red_LED_PIN, OUTPUT);
pinMode(green_LED_PIN, OUTPUT);
pinMode(blue_LED_PIN, OUTPUT);
pinMode(buzzer,  OUTPUT);
pinMode(smoke_detector, INPUT);
Serial.begin(9600);
}
void  loop() {
  int sensor_read = analogRead(smoke_detector);
Serial.print("Smoke  Density: ");
Serial.println(sensor_read);
  if (sensor_read>safety_lim)
  {
          analogWrite(red_LED_PIN,255);
analogWrite(green_LED_PIN, 0);
    tone(buzzer,500, 100);
  }
  else
  {
analogWrite(green_LED_PIN, 255);
analogWrite(red_LED_PIN,0);
noTone(buzzer);
  }
  delay(50);
}
```

OUTPUT:



Q12.Integrating PIR sensor with arduino using tinkercad Simulator and print its value on serial monitor .





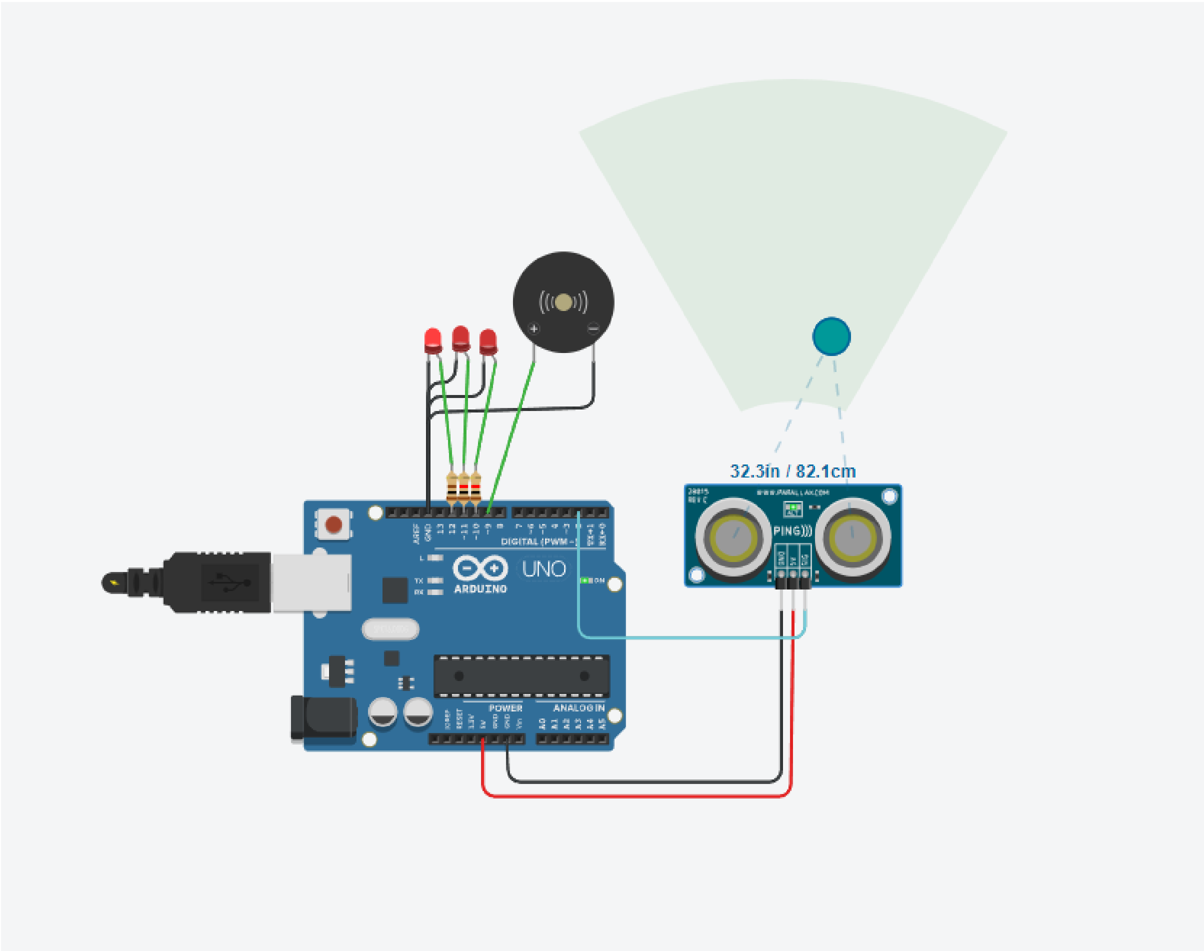
Q13. Measuring the distance using ultrasonic sensors and make LEDs blink using arduino. (Car Parking system).

CODE:

```
int dist = 0;
long readUltrasonicDistance(int triggerPin, int echoPin)
{
pinMode(triggerPin, OUTPUT);  // Clear the trigger
digitalWrite(triggerPin, LOW);
delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
digitalWrite(triggerPin, HIGH);
delayMicroseconds(10);
digitalWrite(triggerPin, LOW);
pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}
void setup()
{
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);
}
void loop()
{
dist = 0.01723 * readUltrasonicDistance(2, 2);
  if (dist> 200) {
digitalWrite(9, LOW);
digitalWrite(10, LOW);
```

```
digitalWrite(11, LOW);
digitalWrite(12, HIGH);
  } else {
    if (dist> 150 &&dist<= 200) {
digitalWrite(9, LOW);
digitalWrite(10, LOW);
digitalWrite(11, HIGH);
digitalWrite(12, HIGH);
    } else {
      if (dist> 100 &&dist<= 150) {
digitalWrite(9, LOW);
digitalWrite(10, HIGH);
digitalWrite(11, HIGH);
digitalWrite(12, HIGH);
      } else {
digitalWrite(9, HIGH);
digitalWrite(10, HIGH);
digitalWrite(11, HIGH);
digitalWrite(12, HIGH);
      }
    }
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```

OUTPUT:



Q14. Home automation system (Sensors: Ultrasonic, PIR, Temperature sensor Actuators: Servomotor, DC-motor, LED's.)
CODE:

```
const int pingPin = 7;
int servoPin = 8;
Servo servo1;
void setup() {
Serial.begin(9600);
  servo1.attach(servoPin);
pinMode(2,INPUT);
pinMode(4,OUTPUT);
pinMode(11,OUTPUT);
pinMode(12,OUTPUT);
pinMode(13,OUTPUT);
pinMode(A0,INPUT);
digitalWrite(2,LOW);
digitalWrite(11,HIGH);
}
void loop() {
  long duration, inches, cm;
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))): a HIGH pulse
  // whose duration is the time (in microseconds) from the sending of the ping
  // to the reception of its echo off of an object.
pinMode(pingPin, INPUT);
```

```
  duration = pulseIn(pingPin, HIGH);
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
  servo1.write(0);
  if(cm < 40)
  {
   servo1.write(90);
   delay(2000);
  }
  else
  {
   servo1.write(0);
  }
   int pir = digitalRead(2);
  if(pir == HIGH)
  {
digitalWrite(4,HIGH);
   delay(1000);
  }
  else if(pir == LOW)
  {
digitalWrite(4,LOW);
  }
  float value=analogRead(A0);
  float temperature=value*0.48;
Serial.println("temperature");
Serial.println(temperature);
  if(temperature > 20)
  {
digitalWrite(12,HIGH);
```

```
digitalWrite(13,LOW);
  }
  else
  {
digitalWrite(12,LOW);
digitalWrite(13,LOW);
  }
}


long microsecondsToInches(long microseconds) {
  return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```

OUTPUT: