Task 4: To interface Ultrasonic with Arduino and write a program for object detection and find the distance between sensor and object.

Aim:

The aim of interfacing an ultrasonic sensor and a buzzer with Arduino is to create a system for object detection and distance measurement. The ultrasonic sensor will be used to measure the distance between the sensor and an object, and if the object is within a certain range (e.g., less than 50 centimeters), a buzzer will be activated to indicate the presence of the object.

Components Required:

1. Arduino Board
2. Bread Board
3. Buzzer
4. Jumper Wires
5. Ultrasonic sensor

Procedure:

Step 1: Connect the Arduino with Ultrasonic sensor using jumper wires.

Step 2: Give the connections from Arduino board to ultrasonic sensor by

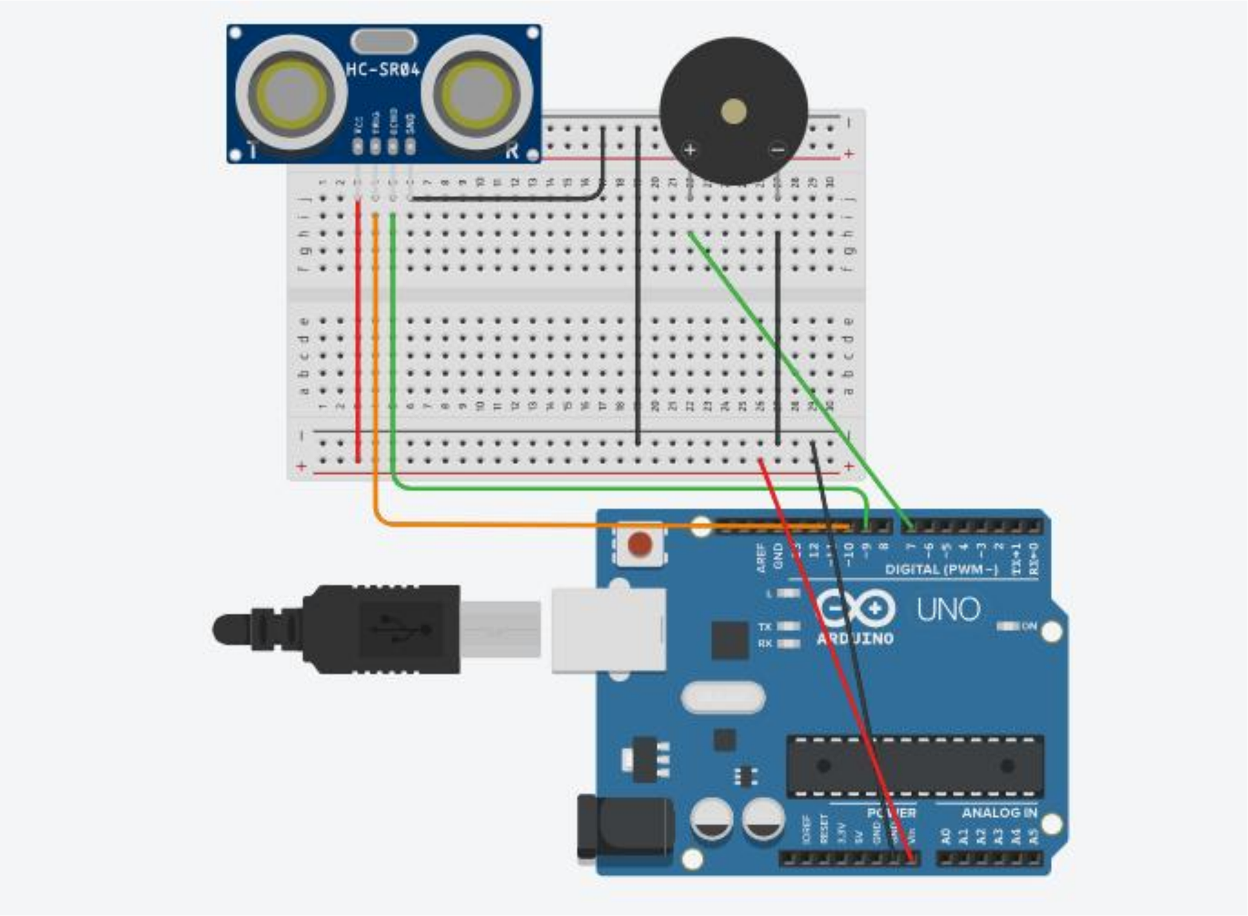
Connect VCC on the ultrasonic sensor to the 5V pin on the Arduino.
Connect the Trig pin on the ultrasonic sensor to pin 9 on the Arduino.
Connect the Echo pin on the ultrasonic sensor to pin 10 on the Arduino.
Connect the Buzzer +ve to pin 11 on the Arduino.
Connect the GND on the ultrasonic sensor to GND on the Arduino.

Step 3: After giving the connections as per schematic diagram, the USB port power cable of Arduino could be connected to PC or laptop.

Step 4: In the laptop Arduino IDE is opened and run the program.

Step 5: After the program is uploaded to Arduino board the ultrasonic sensor is checked whether it is working or not by using serial monitor.

Circuit Diagram:

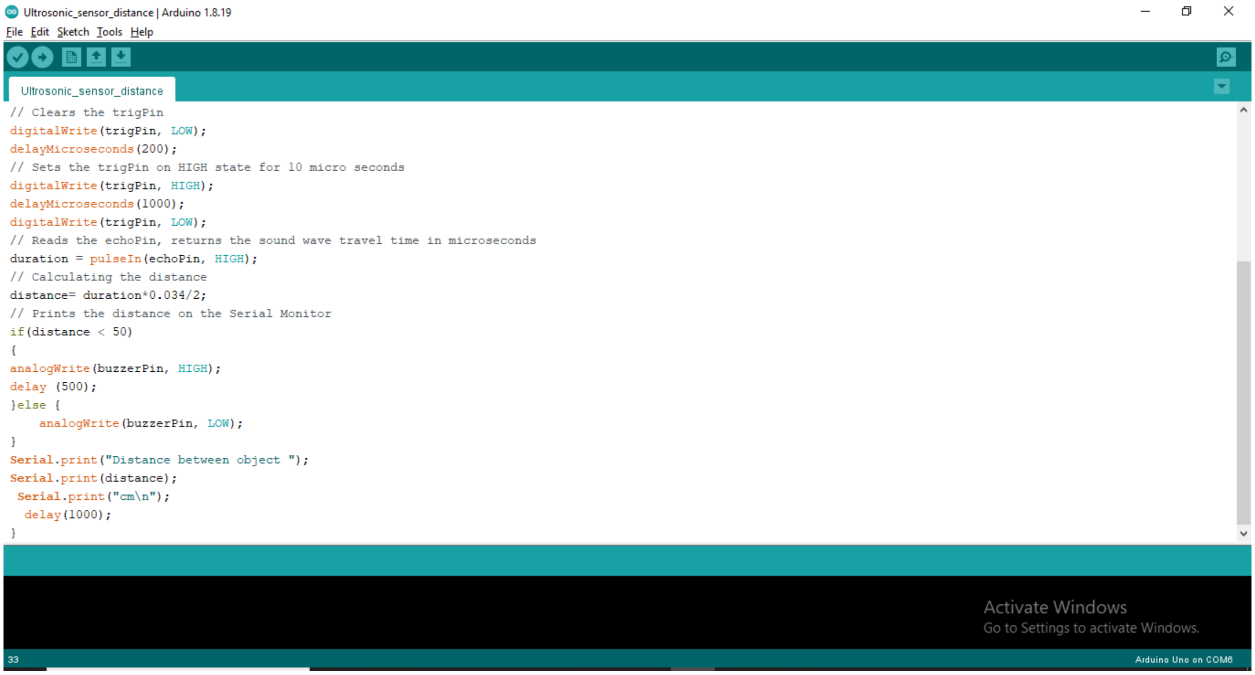


Code:

```
const int trigPin = 9;

const int echoPin = 10;

const int buzzerPin = 11;

int cm;

// defines variables

long duration;
```

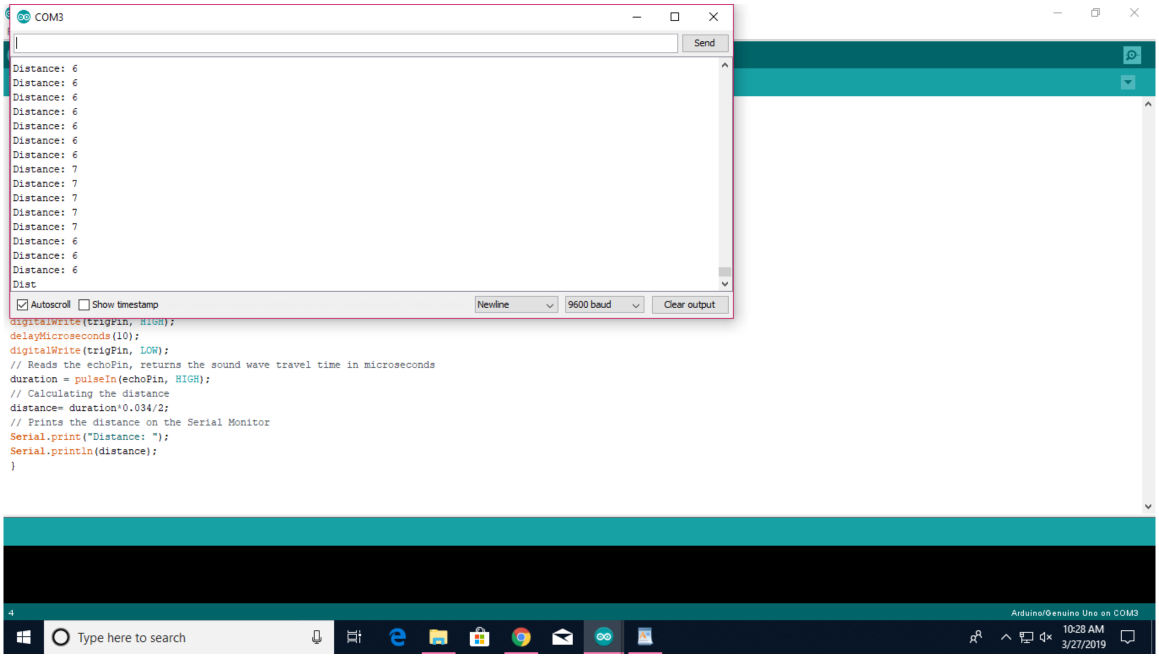
```
int distance;

void setup() {

pinMode(trigPin, OUTPUT);  // Sets the trigPin as an Output

pinMode(echoPin, INPUT);  // Sets the echoPin as an Input

Serial.begin(9600);  // Starts the serial communication

}

void loop() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(200);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(1000);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance= duration*0.034/2;

// Prints the distance on the Serial Monitor

if(distance < 50)

{

analogWrite(buzzerPin, HIGH);

delay (500);

}else {

    analogWrite(buzzerPin, LOW);
}
Serial.print("Distance between object ");
Serial.print(distance);
 Serial.print("cm\n");
  delay(1000);
}
```

Screenshots:



Ultrosonic_sensor_distance | Arduino 1.8.19
File Edit Sketch Tools Help
Ultrosonic_sensor_distance
Activate Windows
Go to Settings to activate Windows.
33
Arduino Uno on COM6

```
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(200);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(1000);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
if(distance < 50)
{
analogWrite(buzzerPin, HIGH);
delay (500);
}else {
    analogWrite(buzzerPin, LOW);
}
Serial.print("Distance between object ");
Serial.print(distance);
 Serial.print("cm\n");
  delay(1000);
}
```



```
Distance: 6
Distance: 6
Distance: 6
Distance: 6
Distance: 6
Distance: 6
Distance: 6
Distance: 7
Distance: 7
Distance: 7
Distance: 7
Distance: 7
Distance: 6
Distance: 6
Distance: 6
Dist
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud Clear output

```
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

Result:

The result of the experiment will be a functioning system that can detect objects within a specified range and provide an audible alert through the buzzer. The Arduino code will continuously measure the distance using the ultrasonic sensor and trigger the buzzer when the measured distance falls below a predefined threshold.