Experiment Name: Distance measurement using Ultrasonic sensor and Arduino

Introduction:

Ultrasonic sensors are widely used for non-contact distance measurement in various applications, including robotics, automation, and obstacle detection. These sensors work by emitting ultrasonic waves and measuring the time taken for the echo to return after reflecting off an object. This time-of-flight is then converted into distance using the speed of sound in air. In this project, an ultrasonic sensor HC-SR04 is interfaced with an Arduino microcontroller to measure distances. The Arduino processes the sensor's data and calculates the distance, which can be displayed on a screen or transmitted to a computer. This setup demonstrates the practical application of ultrasonic technology in measuring distances accurately and efficiently.

Explanation:

An **ultrasonic sensor** is an electronic device used to measure distances by emitting high-frequency sound waves, typically beyond the range of human hearing (ultrasound), and detecting the time it takes for these waves to bounce back after striking an object. These sensors operate by emitting a chirp, usually between 23 kHz and 40 kHz, which is much higher than the audible range of human hearing (up to 20 kHz), hence the term "ultrasonic". The sensor calculates distance using the time-of-flight of the sound wave and the speed of sound in air. This principle is similar to echolocation used by bats to locate their prey. The formula to calculate distance is:

Distance (meters) = (time elapsed [seconds] * 343 [meters/second]) / 2

In this equation, the speed of sound (343 m/s) is used, and the division by 2 accounts for the round trip of the sound wave (to the object and back). The units can be adjusted to fit specific application requirements, making the ultrasonic sensor a versatile tool for precise and efficient distance measurement.

Pins of Ultrasonic:

The typical **ultrasonic sensor**, such as the HC-SR04, has four pins, each with a specific function:

1. **VCC:** This pin powers the sensor. It is connected to the 5V output of the Arduino or another power source.
2. **GND**: This pin is the ground connection, which must be connected to the ground (GND) of the Arduino or circuit.

3. **TRIG**: The trigger pin is used to send an ultrasonic pulse. It requires a short HIGH signal (usually 10 microseconds) to start the measurement.
4. **ECHO**: The echo pin outputs a HIGH signal that lasts for the duration of the time it takes for the ultrasonic pulse to return to the sensor after bouncing off an object. This signal is used to calculate the distance.

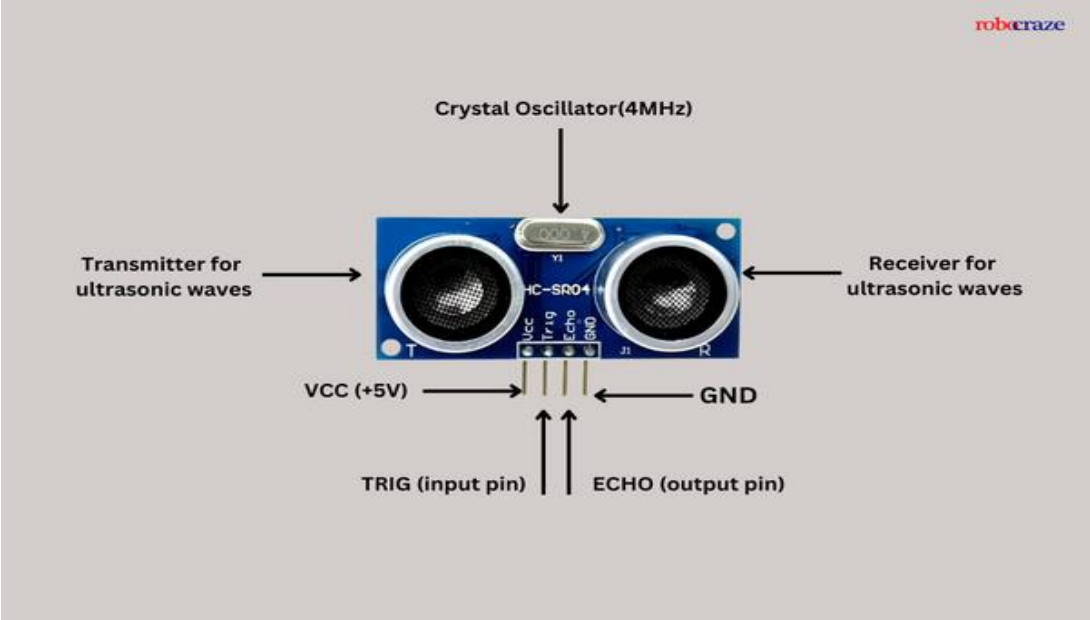

Figure: Ultrasonic (HC-SR04)

Components Required:

1. Arduino Uno R3 board
2. Ultrasonic sensor (HC-SR04)
3. Jumper Wires

Pin Connections:

1. Connect the Echo pin of the sensor to the D2 pin of the Arduino.
2. Connect the Trig pin of the sensor to the D3 pin of the Arduino.
3. Connect the GND and VCC of the sensor to the GND and 5v power pin of the arduino.

Diagram:

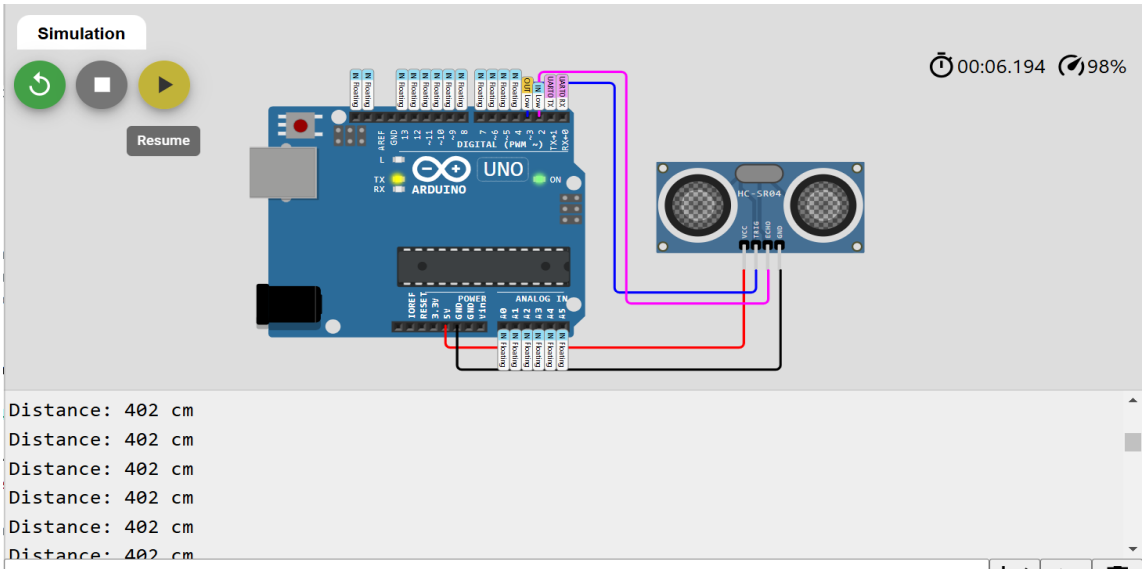

Figure: Circuit Diagram.

Code Snippet:

```
#define echoPin 2
#define trigPin 3
long duration;
int distance;
void setup()
{
   pinMode(trigPin, OUTPUT);
   pinMode(echoPin, INPUT);
   Serial.begin(9600);
   Serial.println("Distance measurement using Arduino Uno.");
   delay(500);
}
void loop()
{
   digitalWrite(trigPin, LOW);
   delayMicroseconds(2);

   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0344 / 2;
 Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(100);
}
```

Code Explanation:

- #define echoPin 2 and #define trigPin 3 assign pin 2 to the Echo pin and pin 3 to the Trigger pin of the ultrasonic sensor. These constants make it easy to identify and change pin connections if needed.
- long duration; stores the time taken by the ultrasonic pulse to travel to the object and back
- int distance; holds the calculated distance based on the pulse duration.
- pinMode(trigPin, OUTPUT); sets the trigger pin as an output to send pulses.
- pinMode(echoPin, INPUT); sets the echo pin as an input to receive reflected pulses.
- Serial.begin(9600); initializes communication between the Arduino and the serial monitor at a baud rate of 9600.
- Serial.println("Distance measurement using Arduino Uno."); displays a message on the serial monitor to indicate that the program has started.
- delay(500); provides a brief pause for the system to stabilize before taking measurements.
- digitalWrite(trigPin, LOW); ensures the trigger pin is initially LOW, stabilizing it before starting a new measurement.
- delayMicroseconds(2); introduces a small pause to prevent errors.
- duration = pulseIn(echoPin, HIGH); measures the time the echo pin stays HIGH, which corresponds to the time taken for the ultrasonic pulse to travel to the object and back.
- distance = duration * 0.0344 / 2; calculates the distance of the object in centimeters using the formula: Distance (meters) = (time elapsed [seconds] * 343 [meters/second]) / 2
- Serial.print("Distance: "); sends the label "Distance: " to the serial monitor.
- Serial.print(distance); sends the calculated distance value to the serial monitor.
- Serial.println(" cm"); adds "cm" to indicate the unit of measurement and moves the cursor to the next line.
- delay(100); pauses for 100 milliseconds before repeating the process to ensure smooth operation.

Conclusion:

The ultrasonic distance measurement system operates by emitting ultrasonic pulses via the Trigger pin, which travel through the air, reflect off an object, and return to the Echo pin. The time taken for this round trip is measured, and the distance is calculated using the speed of sound. The Arduino processes this data and displays the result in centimeters on the serial monitor, providing an accurate, real-time measurement system. This simple yet efficient setup demonstrates the practical application of ultrasonic sensors in distance measurement.