DOP: / /2023 DOS: / / 2023

Experiment No: 03

Arduino - Ultrasonic Sensor

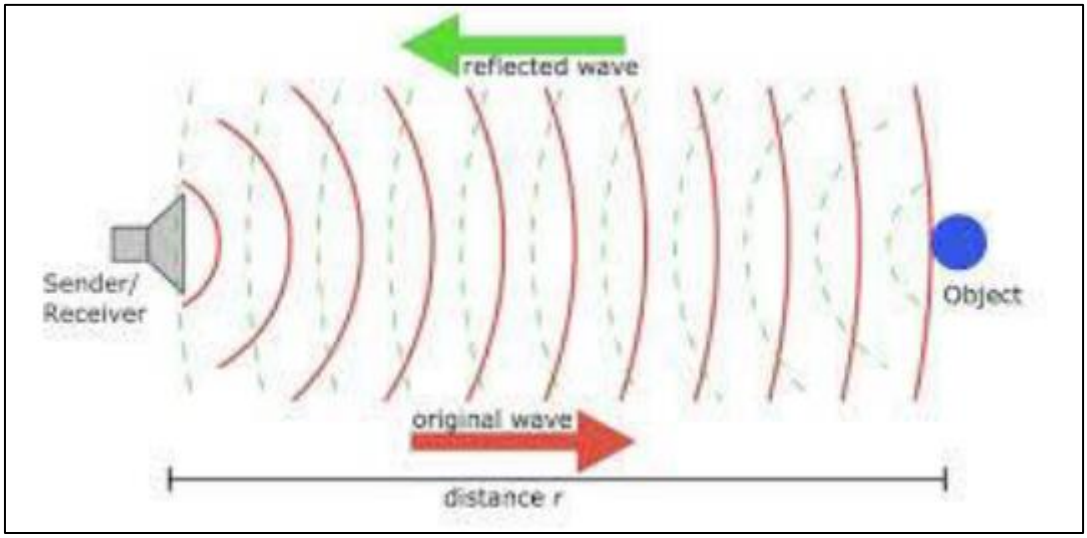
Aim: - Arduino - Ultrasonic Sensor

Theory:

Ultrasonic Sensor:

The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module.



Technical Specifications:

- **Power Supply − +5V DC**
- **Quiescent Current − <2mA**
- **Working Current − 15mA**
- **Effectual Angle − <15°**
- **Ranging Distance − 2cm – 400 cm/1″ – 13ft**
- **Resolution − 0.3 cm**
- **Measuring Angle − 30 degree**

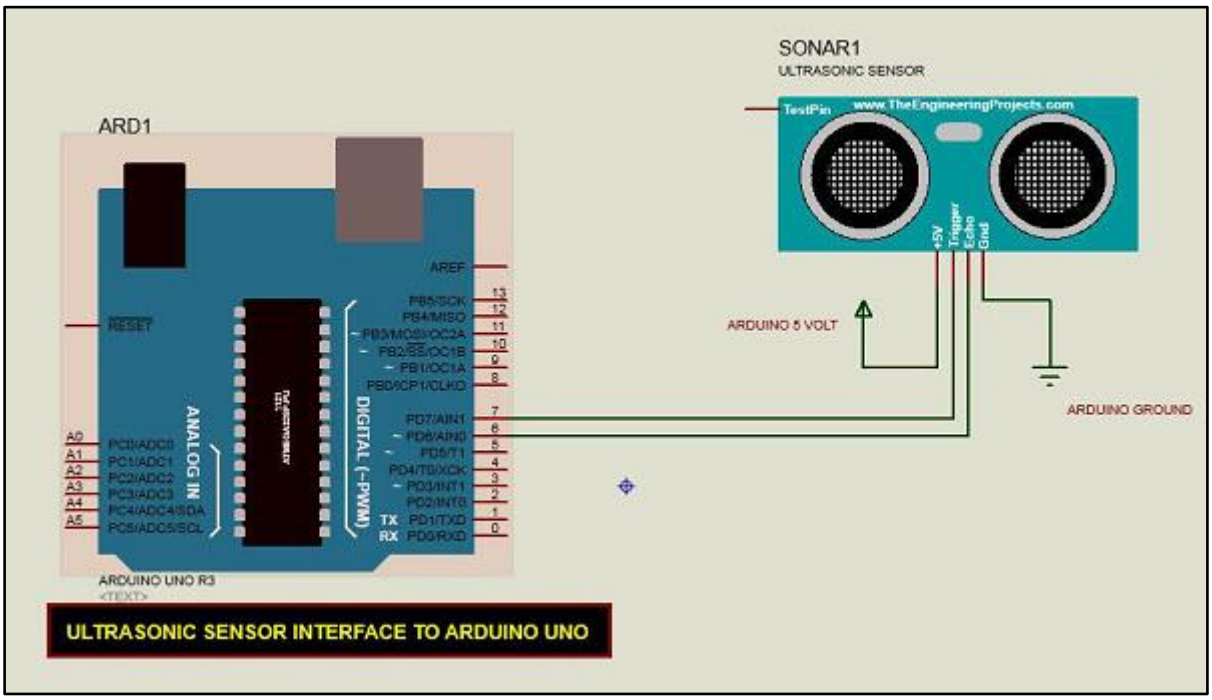
Components Required

You will need the following components −

- **1 × Breadboard**
- **1 × Arduino Uno R3**
- **1 × ULTRASONIC Sensor (HC-SR04)**

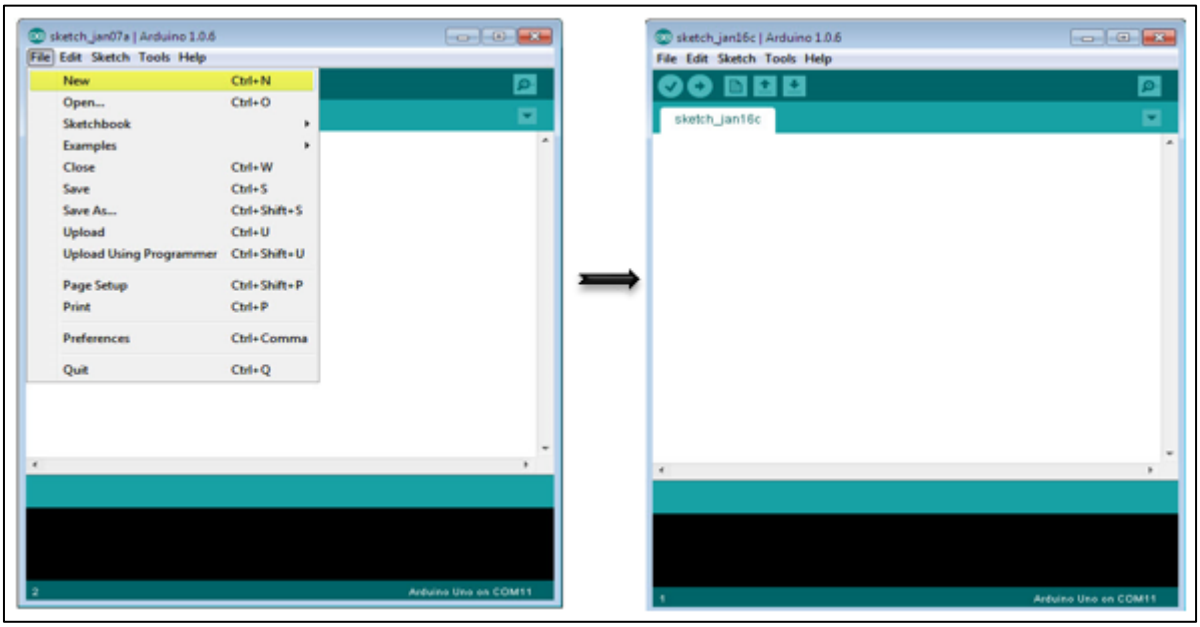
Procedure

Follow the circuit diagram and make the connections as shown in the image given below.



Sketch:

Open the Arduino IDE software on your computer. Coding in the Arduino language will control your circuit. Open a new sketch File by clicking New.



Arduino Code:

```
const int pingPin = 7; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6; // Echo Pin of Ultrasonic Sensor

void setup() {
  Serial.begin(9600); // Starting Serial Terminal
}

void loop() {
  long duration, inches, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(100);
}

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

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

Code to Note:

The Ultrasonic sensor has four terminals - +5V, Trigger, Echo, and GND connected as follows −

- Connect the +5V pin to +5v on your Arduino board.
- Connect Trigger to digital pin 7 on your Arduino board.
- Connect Echo to digital pin 6 on your Arduino board.
- Connect GND with GND on Arduino.

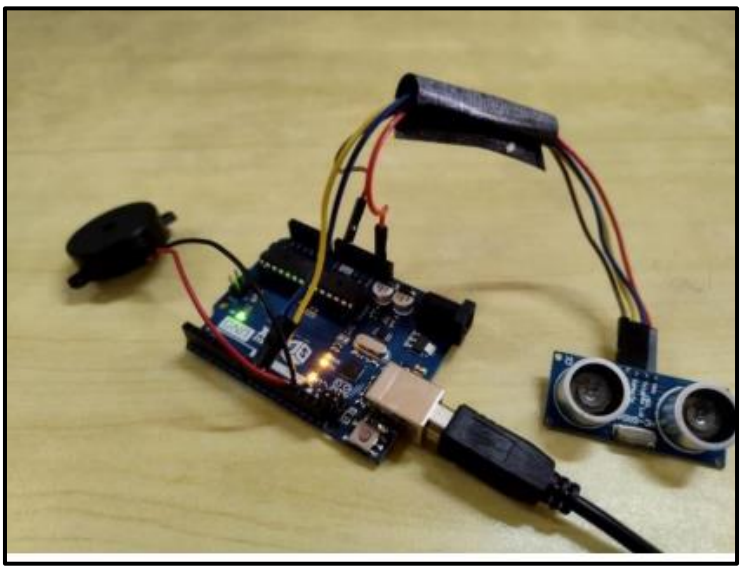
In our program, we have displayed the distance measured by the sensor in inches and cm via the serial port.

Function description:

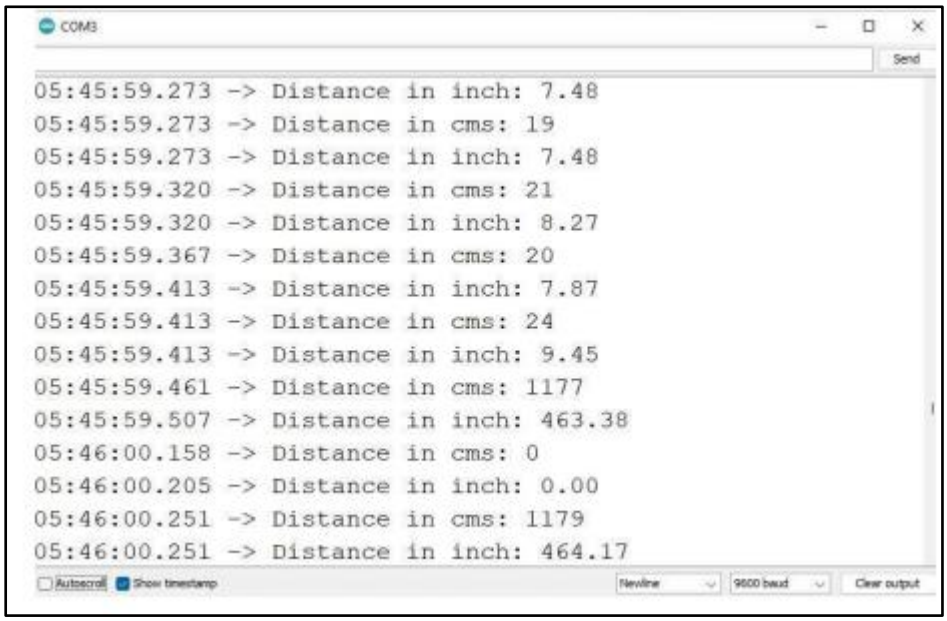
```
long microsecondsToInches(long microseconds)
{
// According to the Ping sensor datasheet, there are
// 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
// second). This gives the distance travelled by the ping, outbound
// and return, so we divide by 2 to get the distance of the obstacle.
return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
// The speed of sound is 340 m/s or 29 microseconds per centimeter.
// The ping travels out and back, so to find the distance of the
// object we take half of the distance travelled.
return microseconds / 29 / 2;
}
```

Result:

You will see the distance measured by sensor in inches and cm on Arduino serial monitor. And when the distance measured by the sensor is 20 cm or less, the buzzer will turn on, otherwise it will remain off.



Output:



Conclusion:

The Arduino ultrasonic sensor is a versatile and cost-effective tool for measuring distance, proximity and obstacle detection in various applications. It operates by sending and receiving high-frequency sound waves and measuring the time delay between the transmission and reception. The sensor provides a simple interface to read the distance data, making it easy to integrate into various projects. With its small size, low power consumption, and high accuracy, the Arduino ultrasonic sensor is a popular choice for DIY projects and educational purposes.

1) Connected the VCC pin of the sensor to 5V power supply on the Arduino board.
2) Connected the GND pin of the sensor to GND on the Arduino board.
3) Connected the Trig pin of the sensor to a digital pin on the Arduino board (e.g., pin 9).
4) Connected the Echo pin of the sensor to another digital pin on the Arduino board (e.g., pin 8).
5) Connected the positive terminal of the buzzer to a digital pin on the Arduino board (e.g., pin 13).
6) Connected the negative terminal of the buzzer to GND on the Arduino board. When the distance measured by the sensor is 20 cm or less, the buzzer will turn on, otherwise it will remain off.