



Distance measurement using Ultrasonic sensor and Arduino

Last Updated : 30 Apr, 2024

In This Article, We will learn how to measure the distance using an Ultrasonic sensor & Arduino.

Arduino:

It is an open-source electronics platform. It consists ATmega328 8-bit Microcontroller. It can be able to read inputs from different sensors & we can send instructions to the microcontroller in the Arduino. It provides Arduino IDE to write code & connect the hardware devices like Arduino boards & sensors.

Ultrasonic Sensor:

An ultrasonic Sensor is a device used to measure the distance between the sensor and an object without physical contact. This device works based on time-to-distance conversion.

Working Principle of Ultrasonic Sensor:

Ultrasonic sensors measure distance by sending and receiving the ultrasonic wave. The ultrasonic sensor has a sender to emit the ultrasonic waves and a receiver to receive the ultrasonic waves. The transmitted ultrasonic wave travels through the air and is reflected by hitting the Object. Arduino calculates the time taken by the ultrasonic pulse wave to reach the receiver from the sender.

We know that the speed of sound in air is nearly 344 m/s,

So, the known parameters are time and speed (constant). Using these parameters, we can calculate the distance traveled by the sound wave.

In the code, the “duration” variable stores the time taken by the sound wave traveling from the emitter to the receiver. That is double the time to reach the object, whereas the sensor returns the total time including sender to object and object to receiver. Then, the time taken to reach the object is half of the time taken to reach the receiver.

so we can write the expression as,

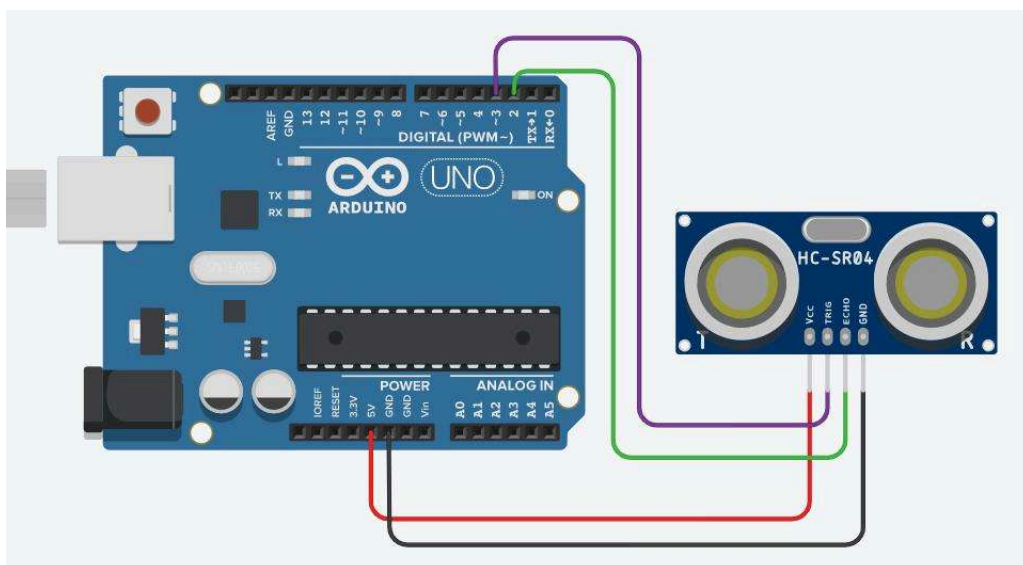
$$\text{Distance} = \text{Speed of Sound in Air} * (\text{Time Taken} / 2)$$

Note: Speed of sound in air = 344 m/s.

Components Required:

1. Arduino Uno R3 board
2. Ultrasonic sensor (HC-SR04)
3. 16×2 LCD I2C Display
4. Jumper Wires

Circuit Diagram:



Ultrasonic Distance measurement circuit

2. Connect the Trig pin of the sensor to the D3 pin of the Arduino.
3. Navigate to Tools and select board and port.
4. Verify and compile the code, then upload the code to the Arduino Uno R3 board.
5. Monitor the output in the Serial monitor (Set the baud rate as 9600). To open Serial monitor **Tools>Serial Monitor** or (**Ctrl+Shift+M**).

Arduino Code (Output in Serial monitor):

The following code will show the output on the serial monitor of Arduino software with a baud rate of 9600.

```
#define echoPin          \
    2 // attach pin D2 Arduino to pin Echo of HC-SR04          \
#define trigPin          \
    3 // attach pin D3 Arduino to pin Trig of HC-SR04          \
\

long duration; // Variable to store time taken to the pulse
                // to reach receiver

int distance; // Variable to store distance calculated using
              // formula

void setup()
{
    pinMode(trigPin,
              OUTPUT); // Sets the trigPin as an OUTPUT
    pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

    // Serial Communication is starting with 9600 of
    // baudrate speed
    Serial.begin(9600);

    // The text to be printed in serial monitor
    Serial.println(
        "Distance measurement using Arduino Uno.");
    delay(500);
}

void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2); // wait for 2 ms to avoid
                          // collision in serial monitor

    digitalWrite(
        trigPin,
        HIGH); // turn on the Trigger to generate pulse
    delayMicroseconds(
        10); // keep the trigger "ON" for 10 ms to generate
```

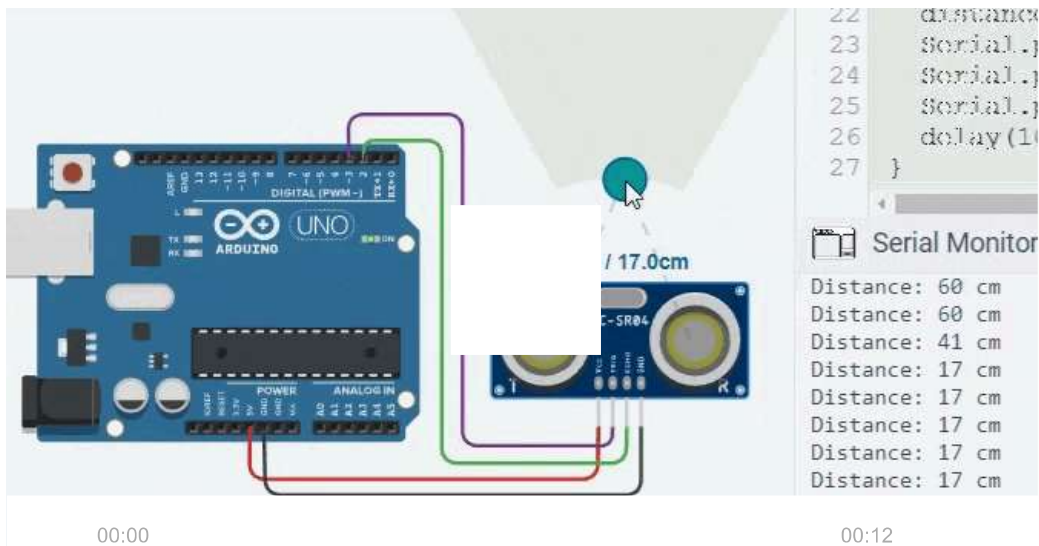
```

// If pulse reached the receiver echoPin
// become high Then pulseIn() returns the
// time taken by the pulse to reach the
// receiver

duration = pulseIn(echoPin, HIGH);
distance
    = duration * 0.0344 / 2; // Expression to calculate
                             // distance using time

Serial.print("Distance: ");
Serial.print(
    distance); // Print the output in serial monitor
Serial.println(" cm");
delay(100);
}

```



Output in External Display device (LCD Display):

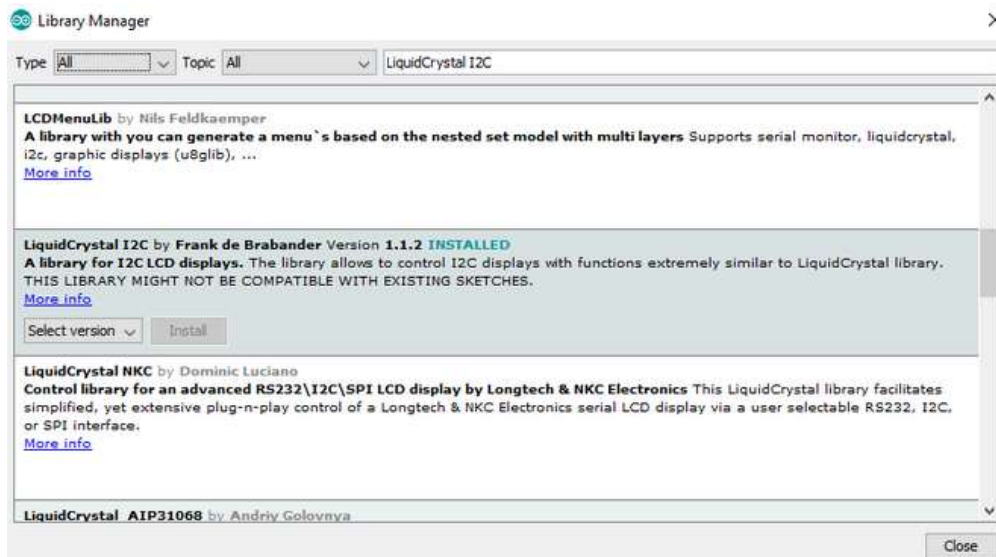
To monitor the output without a PC, we have to interface display devices to monitor the output. Here, we are using “a 16×2 LCD display with I2C communication.

To interface with the LCD display, we need to install the supporting library to the Arduino IDE.

Steps to Interface LCD display:

1. Install driver library for Liquid Crystal Display.

- Search for “LiquidCrystal I2C” and install the “LiquidCrystal I2C” library.



Library Installation

2. Import the header file “LiquidCrystal_I2C.h” in the code.
3. Connect the SDA pin of an LCD display to the SDA pin of the Arduino Board and the SCL pin of an LCD display to the SCL of the Arduino Board.
4. Connect VCC to 5V pin and GND to GND pin.
5. Include the below code to define the display device.

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C
    lcd(0x20, 16, 2); // Format => (ADDRESS,Width,Height )

void setup()
{
    lcd.init(); // Initialize the lcd
    lcd.backlight(); // Turn on the Backlight

    // ....
}
```

Note: For ADDRESS, Try “0x27” or “0x3F”. If those addresses are not working. Refer manufacturer page.

Or

To find the address, Run the following code to find the address.

Arduino code for I2C device Address Finding:

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

```

void setup()
{
    Wire.begin();
    Serial.begin(9600);
    Serial.println("I2C Scanner");
}

void loop()
{
    byte error, address;
    int nDevices;

    Serial.println("Scanning...");

    nDevices = 0;
    for (address = 1; address < 127; address++) {
        Wire.beginTransmission(address);
        error = Wire.endTransmission();

        if (error == 0) {
            Serial.print("I2C device found at address 0x");
            if (address < 16)
                Serial.print("0");

            Serial.print(address, HEX);
            Serial.println(" !");

            nDevices++;
        }
        else if (error == 4) {
            Serial.print("Unknown error at address 0x");
            if (address < 16)
                Serial.print("0");

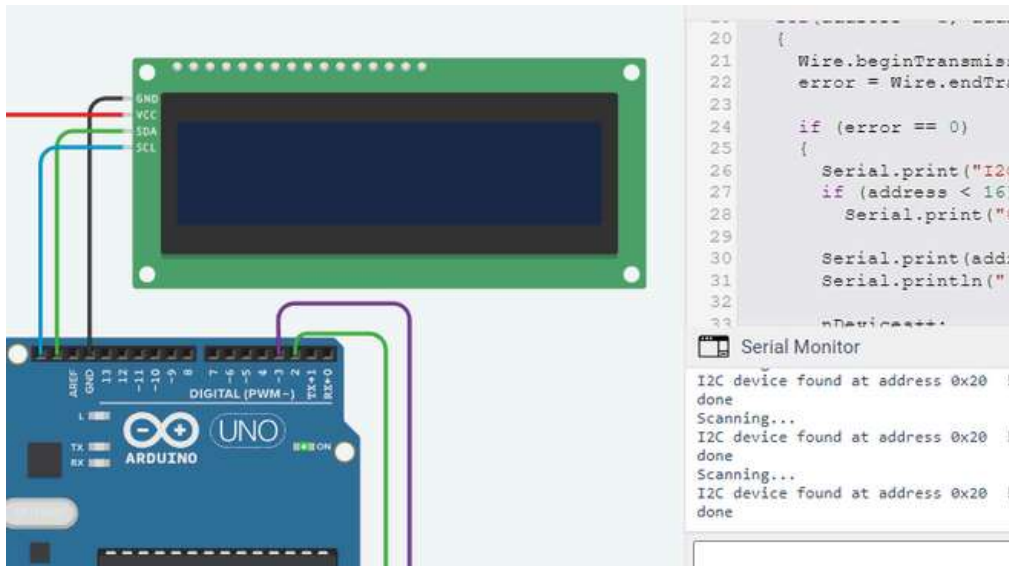
            Serial.println(address, HEX);
        }
    }

    if (nDevices == 0)
        Serial.println("No I2C devices found");
    else
        Serial.println("done");

    delay(5000); // wait 5 seconds for next scan
}

```

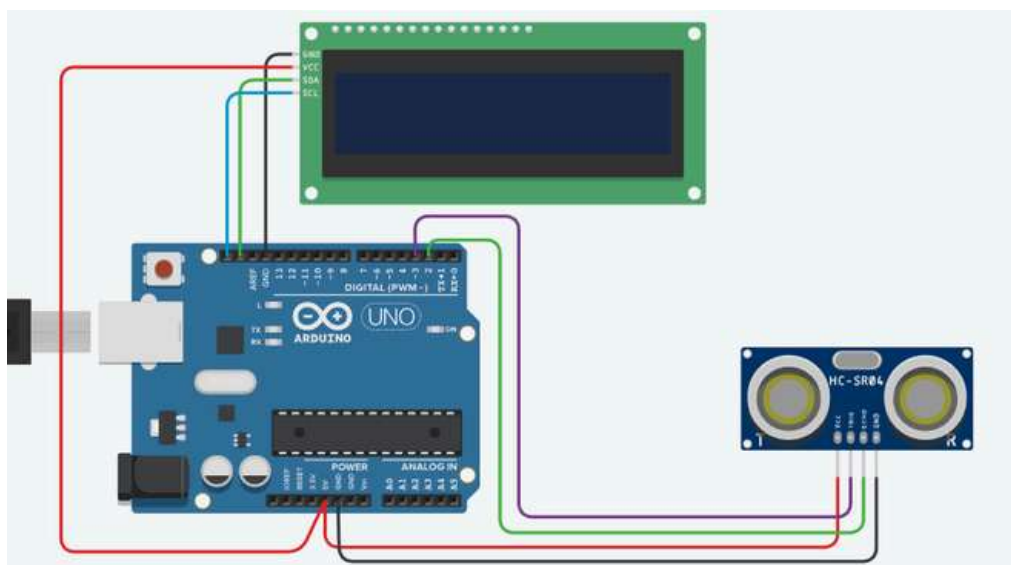
Serial monitor Output:



LCD Address Finding

Here I2C Address is “0x20”, So replace the address with “0x20” in the code.
(Ensure that you have connected the LCD display device as per the instruction while running this program in the Arduino).

Circuit Diagram with LCD display:



Ultrasonic Distance Measurement Output in LCD Display

Arduino code (Output in Serial Monitor + LCD display):

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x20, 16, 2); // Format -> (Address,Width,Height )

#define echoPin 2 // attach pin D2 Arduino to Echo pin of Sensor module
#define trigPin 3 // attach pin D3 Arduino to Trig pin of Sensor module

long duration; // Declare variable to store echo time duration
```

```

{
  lcd.init(); // initialize the lcd
  lcd.backlight(); // Turn on the Backlight

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

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

  // Serial Communication is starting with 9600 of baudrate speed
  Serial.begin(9600);

  // The text to be printed in serial monitor
  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");

  lcd.clear(); // Clear the display buffer
  lcd.setCursor(0, 0); // Set cursor for "Distance:" (Column, Row)

  lcd.print("Distance:"); // print "Distance:" at (0, 0)
  lcd.setCursor(0,1); // Set cursor for output value (0, 1)

  lcd.print(distance); // print Output in cm at (0, 1)
  lcd.setCursor(4, 1); // move cursor to (4, 1)
  lcd.print("cm"); // print "cm" at (4, 1)

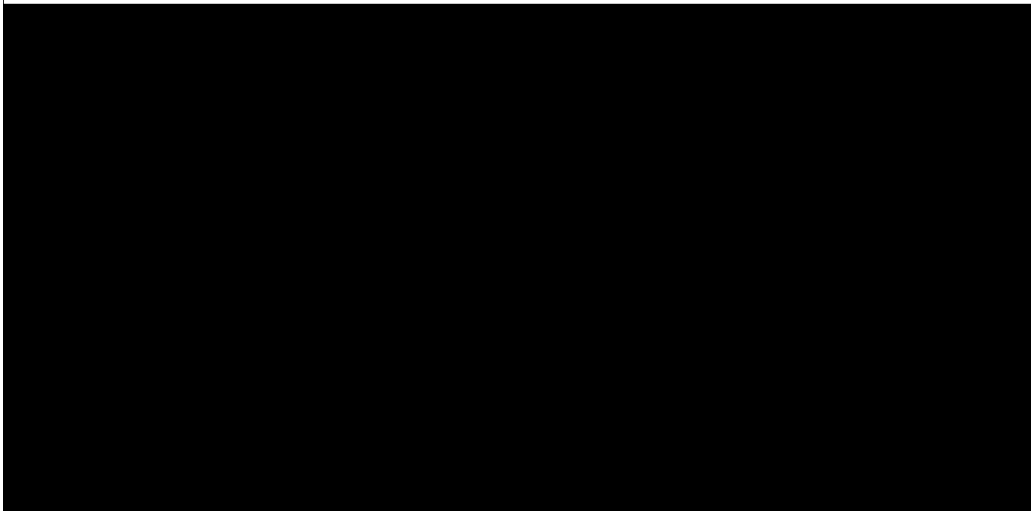
  delay(100);
}

```

Output:

00:00

00:08



Comment

More info

Advertise with us

Next Article

Arduino - Ultrasonic Sensor

Similar Reads

Distance measurement using Ultrasonic sensor and Arduino

In This Article, We will learn how to measure the distance using an Ultrasonic sensor & Arduino. Arduino:It is an open-source electronics platform. It consists ATmega328 8-bit Microcontroller. It can be able to rea...

7 min read

Arduino - Ultrasonic Sensor

Sensors are widely used for detecting devices by approximating their distance from the source. One such example of a sensor is the HC-SR04 ultrasonic sensor which uses the SONAR technique for the sensing...

7 min read

Ultrasonic Sensor Vs IR Sensor

Arduino is a microcontroller that requires sensors to sense the surroundings to process and react to it. Ultrasonic Sensor is an electronic device that measures the distance between the transmitter and the...

6 min read

Servo motor Interfacing and Control using Arduino

In this article, we will learn how to interface and control servo motors using Arduino Uno R3. Arduino is an open-source electronics platform. It consists ATmega328P 8-bit Microcontroller. It can be able to read...

3 min read

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Proximity Sensor is one of the sensors present in mobile devices which we use almost every day. This sensor is present in the top section of your phone. The sensor is used to detect the presence of any object...

3 min read

How to make Motion Detection System using Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read digital & analog inputs from the sensors and The PIR sensor is a special type of sensor...

2 min read

Euclidean Distance using Scikit-Learn - Python

Scikit-Learn is the most powerful and useful library for machine learning in Python. It contains a lot of tools, that are helpful in machine learning like regression, classification, clustering, etc. Euclidean distanc...

3 min read

Smart Collision Detector using Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read digital & analog inputs from the sensors and the HR SC-04 Ultrasonic sensor uses sonar...

4 min read

Arduino - Water Detector / Sensor

Sensors are widely used for detecting different quantities like the presence of light, level of water, etc. One such example of a sensor is the water sensor which uses various principles of physics to detect the...

8 min read

Minimum distance from a point to the line segment using Vectors

Given the coordinates of two endpoints $A(x_1, y_1)$, $B(x_2, y_2)$ of the line segment and coordinates of a point $E(x, y)$; the task is to find the minimum distance from the point to line segment formed with the given...

10 min read



Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).



Advertise with us

Company

About Us
Legal
Privacy Policy
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved