**Lab Exercise 10- Ultrasonic Sensor with Arduino and Displaying Data on Serial Monitor**

In this lab exercise, you will learn how to use an **HC-SR04 ultrasonic sensor** to measure the distance of an object and display the result on the **Serial Monitor** using an Arduino.

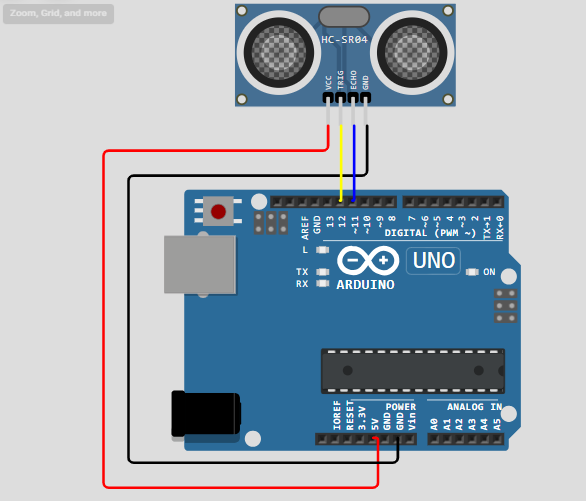
**Components Required:**

* 1 x Arduino Uno
* 1 x Ultrasonic Sensor (HC-SR04)
* Jumper wires
* Breadboard (optional)

**Circuit Diagram:**

**Ultrasonic Sensor Pin Configuration:**

1. **VCC**: Connect to **5V** of the Arduino.
2. **GND**: Connect to **GND** of the Arduino.
3. **Trig**: Connect to **Digital Pin 12** of the Arduino.
4. **Echo**: Connect to **Digital Pin 11** of the Arduino.



**Working Principle of HC-SR04:**

* The **HC-SR04** sensor emits an ultrasonic pulse (sound wave) when the **Trig** pin is triggered.
* It waits for the pulse to reflect back from an object, and the time it takes for the echo to return is used to calculate the distance.

**Arduino Code:**

// Define pin numbers

const int trigPin = 12; // Pin connected to Trig of HC-SR04

const int echoPin = 11; // Pin connected to Echo of HC-SR04

// Variables to store time and distance

long duration;

float distance;

void setup() {

// Start the serial communication

Serial.begin(9600);

// Define the pin modes

pinMode(trigPin, OUTPUT); // Trig pin as output

pinMode(echoPin, INPUT); // Echo pin as input

Serial.println("Ultrasonic Sensor HC-SR04 Initialized");

}

void loop() {

// Clear the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Trigger the sensor by setting the trigPin HIGH for 10 microseconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Read the echoPin, and calculate the duration it took for the pulse to return

duration = pulseIn(echoPin, HIGH);

// Calculate the distance in cm (duration \* 0.034 / 2)

distance = (duration \* 0.034) / 2;

// Print the distance to the Serial Monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Wait before the next measurement

delay(500);

}

**Explanation of Code:**

1. **Pin Setup**:
   * The **trigPin** is connected to digital pin 9 and is used to trigger the sensor to send out a pulse.
   * The **echoPin** is connected to digital pin 10 and receives the echo pulse.
2. **Setup Function**:
   * Serial.begin(9600) initializes serial communication to display data on the Serial Monitor.
   * pinMode(trigPin, OUTPUT) sets the Trig pin as output to send the ultrasonic pulse.
   * pinMode(echoPin, INPUT) sets the Echo pin as input to receive the reflected pulse.
3. **Loop Function**:
   * The **Trig pin** is set LOW and then HIGH for **10 microseconds** to generate an ultrasonic pulse.
   * The pulseIn() function measures the time taken for the pulse to be received by the **Echo pin**.
   * The distance is calculated using the formula: distance (cm)=duration×0.0342\text{distance (cm)} = \frac{\text{duration} \times 0.034}{2}distance (cm)=2duration×0.034​
     + Where **0.034 cm/µs** is the speed of sound, and the division by 2 accounts for the round-trip time.
   * The distance is printed on the Serial Monitor every 500 milliseconds.

**Uploading the Code:**

1. Connect the Arduino to your computer using a USB cable.
2. Open the Arduino IDE and paste the code into the editor.
3. Select the correct **Board** and **Port** from the "Tools" menu.
4. Click the **Upload** button to upload the code to the Arduino.

**Viewing the Results:**

1. After uploading the code, open the **Serial Monitor** from the Arduino IDE (Tools > Serial Monitor).
2. Set the baud rate to **9600**.
3. You should see the distance between the ultrasonic sensor and the object printed on the Serial Monitor, updated every 500 milliseconds.

**Expected Output on the Serial Monitor:**

Ultrasonic Sensor HC-SR04 Initialized

Distance: 35.42 cm

Distance: 34.78 cm

Distance: 34.91 cm

...

**Troubleshooting Tips:**

* Ensure that the **Trig** and **Echo** pins are properly connected.
* Check the **power supply** to the sensor. The HC-SR04 sensor requires a 5V power supply.
* Ensure there are no obstructions or interference in front of the sensor while measuring.

**Further Modifications:**

* Adjust the **delay()** in the loop to change how often the distance is measured.
* Use **Serial Plotter** (from the Tools menu) to visually see the changes in distance over time.
* Implement **buzzer or LED indicators** to notify when an object is within a certain distance.

**Conclusion:**

In this lab, you successfully interfaced an **HC-SR04 ultrasonic sensor** with an Arduino to measure the distance of objects. The distance is calculated and printed on the **Serial Monitor**, which can be used in various applications such as object detection, obstacle avoidance in robotics, or distance measurement in IoT systems.

**Further Exploration:**

* Combine the ultrasonic sensor with a **servo motor** to create a **distance-measuring radar**.
* Use the sensor to detect objects and trigger an alarm or **buzzer** if an object is within a certain range.
* Integrate the system with an **LCD display** to show the measured distance instead of using the Serial Monitor.