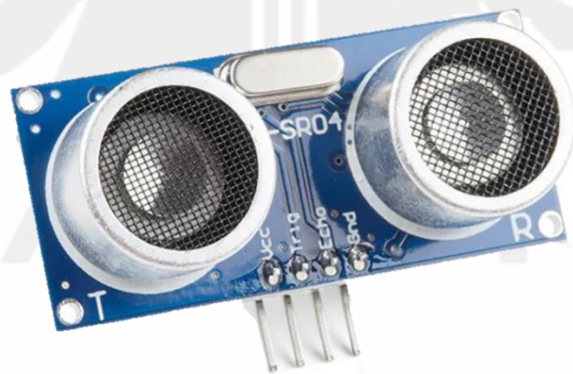


LAUNCH THE PROJECT - 15

Ultrasonic Distance Meter

(using HC-SR04 Sensor)

On AYNOP® UNO Launchpad Kit



Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 22-9-2025

Table of Contents

1. Overview.....	3
2. Components Required.....	3
3. Software Required.....	3
4. Hardware Setup	4
4.1 Wiring Diagram.....	4
5. Principle – How It Works	5
6. Procedure – Steps to Run	6
6.1 How to Use the Serial Monitor:.....	7
7. Expected Output.....	7
8. Code.....	8
8.1 Function References.....	8
9. Troubleshooting Tips.....	9
10. License.....	10
11. Support & Feedback.....	10

1. Overview

This project demonstrates how to use the **HC-SR04 ultrasonic sensor** with Arduino® UNO R4 Minima to create a simple **Ultrasonic Distance Meter**.

You will learn to:

- Connect the HC-SR04 sensor to Arduino.
- Trigger ultrasonic pulses and measure their echoes.
- Calculate distances in centimetres using the time of flight of sound waves.
- Improve accuracy by retrying up to three times and using the first valid reading.

2. Components Required

- Arduino® UNO R4 Minima board
- USB Type-C data cable
- HC-SR04 Ultrasonic Sensor
- Breadboard
- Jumper wires

3. Software Required

- Arduino IDE (v2.3.6 or later recommended)
- No additional libraries required.

Note:

We assume the **Arduino UNO R4 Minima board package** is already installed on your machine, as explained in the 00_Getting_Started/00_GettingStarted_Arduino_R4_Minima guide. If it is not installed, please refer to that document and complete the installation before proceeding.

4. Hardware Setup

This section explains how to connect the HC-SR04 ultrasonic sensor to Arduino.

4.1 Wiring Diagram

- HC-SR04 VCC → 5V.
- HC-SR04 GND → GND.
- HC-SR04 TRIG → D9.
- HC-SR04 ECHO → D10

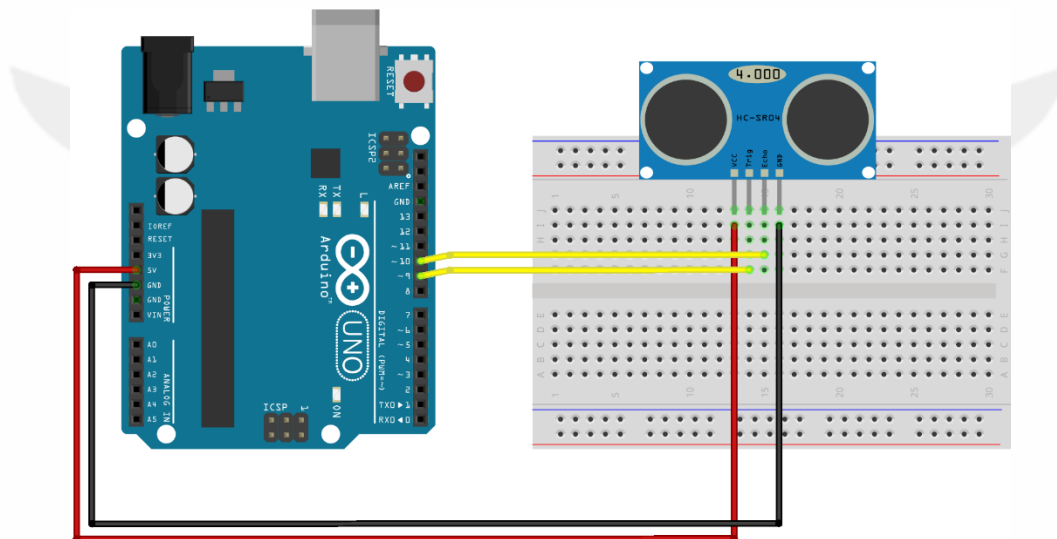


Figure 4.1 – Wiring diagram for Ultrasonic Distance Meter project.

Tip-1: Always disconnect the USB cable before making or changing hardware connections.

Tip-2: Keep the sensor facing forward in an open area. Large, flat objects (like books or walls) give the best reflections.

5. Principle – How It Works

The **HC-SR04 ultrasonic sensor** measures distance by sending out an ultrasonic pulse and listening for the echo. Arduino then calculates the time difference to estimate how far away an object is.

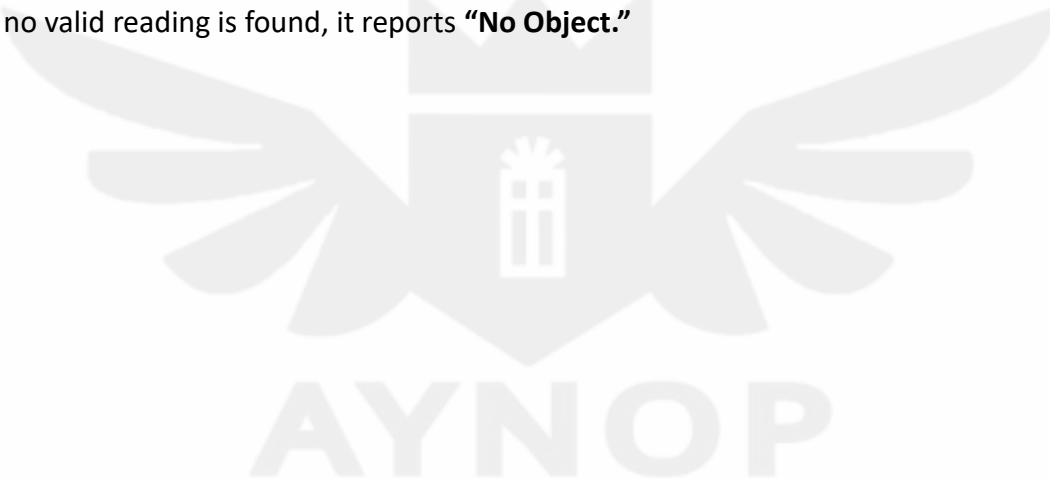
- The **TRIG** pin sends a short 10 μs HIGH pulse to start the measurement.
- The **ECHO pin** goes HIGH for the duration of the echo's return trip.
- The travel time of sound is converted into distance using the formula:

$$\text{Distance (cm)} = (\text{Time in } \mu\text{s} \times 0.034) / 2$$

where 0.034 cm/ μs is the speed of sound, and we divide by 2 since the pulse travels to the object and back.

- The reliable range of HC-SR04 is **2 cm to 400 cm**.

In this project, the Arduino takes up to **3 attempts** for each measurement and uses the first valid result. If no valid reading is found, it reports **"No Object."**



6. Procedure – Steps to Run

1. Build the Circuit

- a. Connect the HC-SR04 as shown in the **Wiring Diagram (Figure 4.1)**.

2. Connect the Board

- a. Use a USB Type-C data cable to connect your UNO R4 Minima to your computer.

3. Open the Project Code

- a. Simply **double-click** the file *15_Ultrasonic_Distance_Meter.ino* in the project folder, and it will open directly in the Arduino IDE (if installed).

4. Confirm Board Selection

- a. The IDE usually auto-detects the UNO R4 Minima if the package is installed.
- b. If not installed, refer to the *00_Getting_Started/00_GettingStarted_Arduino_R4_Minima* document to install the necessary board package.
- c. Verify that *Arduino UNO R4 Minima* is selected in the IDE's board selector (top toolbar).

5. Upload the Code

- a. Click the **Upload** button (arrow icon) in the top-left corner of the IDE.
- b. Wait until the console displays **"Done uploading."**

6. Observe the Behaviour

- a. Place an object (like your hand, book, or box) in front of the sensor.
- b. The Serial Monitor will display the distance in centimetres.
- c. Move the object closer and watch the values change.
- d. If nothing is detected, you will see **"No Object."**

6.1 How to Use the Serial Monitor:

The Serial Monitor is a **tool in the Arduino IDE** that lets you see text/data sent by the microcontroller (MCU) over the USB connection.

- In Arduino IDE, go to **Tools → Serial Monitor**, or press **Ctrl + Shift + M**.
- A new window will open showing real-time data from your UNO R4 Minima.
- At the bottom-right of the Serial Monitor window, set the baud rate to **9600**. This must match the *Serial.begin(9600)* in the code.

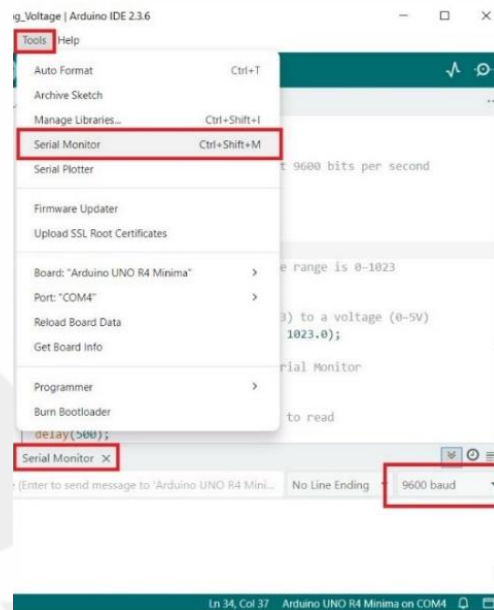


Figure 6.1 – Arduino IDE Serial Monitor set to 9600 baud

Tip: If the Serial Monitor shows nothing, check that:

- The correct **COM port** is selected.
- The baud rate is set to **9600**


7. Expected Output

- The Serial Monitor will show values like:


```
AYNOP: Ultrasonic Distance Meter
Object Distance from Sensor: 96.8 cm
Object Distance from Sensor: 96.9 cm
Object Distance from Sensor: 96.9 cm
```

8. Code

The source code for this project is included in the downloaded folder:

 `uno-launchpad-kit/01_Basic_Projects/15_Ultrasonic_Distance_Meter/15_Ultrasonic_Distance_Meter.ino`


Tip:

- To open the project, simply **double-click the .ino file**. If the Arduino IDE is installed, it will launch automatically and load the code.
- If you **haven't installed the Arduino IDE yet**, please refer to:
 `uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima` to **download and install it**.


8.1 Function References


- `setup()` – runs once when the board is powered on or reset.
- `loop()` – runs continuously after `setup()` finishes.
- `pinMode(pin, mode)` – sets a pin as input or output.
- `digitalWrite(pin, value)` – sets a pin HIGH or LOW.
- `delay(ms)` – pauses program execution for a set time.
- `delayMicroseconds(us)` – pauses program for microseconds.
- `pulseIn(pin, value, timeout)` – measures the duration of a HIGH or LOW pulse.
- `Serial.begin(baud)` – starts serial communication.
- `Serial.print()` / `Serial.println()` – prints values to the Serial Monitor.

 For more details and advanced usage, visit:

 [Arduino Language Reference](#) — The official guide for all Arduino functions.

9. Troubleshooting Tips

- **Always getting “No Object”?**
 - Check that TRIG is wired to pin 9 and ECHO to pin 10.
 - Ensure the object is within the valid range (2–400 cm).
 - Flat, hard surfaces reflect sound better than soft ones.
- **Distance values are unstable?**
 - Avoid very close or angled objects.
 - Hold the sensor steady
 - Ensure no interference from nearby objects.
- **Upload error in Arduino IDE?**
 - Verify that the correct board (**Arduino UNO R4 Minima**) is selected in the IDE.
 - Check that the correct **COM port** is chosen.
- **Board not detected via USB?**
 - Ensure you are using a **data-capable USB Type-C cable** (some cables only provide charging).
 - Try reconnecting the cable or using a different USB port.
- **Board not listed in Arduino IDE?**
 - If you don't see **Arduino UNO R4 Minima** in the board selector, the **board package is not installed**.
 - To fix this, follow the installation steps in:
 [uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima](#)

 **Tip:** If nothing works, press the **RESET** button on the UNO R4 Minima and try uploading the code again.

10. License

This content (source code and documentation) is licensed under **Creative Commons BY-NC-SA 4.0**.
© 2025 AYNOP. You may use, modify, and share for personal and educational purposes only.
Commercial use or redistribution without prior written permission is strictly prohibited.
Refer to the **LICENSE** file for complete details.

11. Support & Feedback

We value your feedback and are happy to assist with any questions, troubleshooting, or suggestions you may have.

✉ Email: support@aynop.com

When sending an email, please include your kit name (AYNOP® UNO Launchpad Kit) and, if applicable, the project name in the subject line. This will help our team respond faster and more accurately. We aim to respond to all queries within 2–3 business days. Your feedback helps us improve our products and create even better learning experiences for all Arduino beginners.

