Lab Report: Arduino Sensor & LCD Interfacing

V.L.S. Bhargav ; Lanka Kushwanth
Table Number: 3
Room Number: 114
Roll Number: 2025102061

October 12, 2025

PART A:

Aim:

The aim of this experiment is to understand the basic working of Arduino UNO by interfacing it with an HC-SR04 Ultrasonic sensor and observing distance measurements on the Serial Monitor.

Components Required:

- Arduino UNO board
- HC-SR04 Ultrasonic sensor
- Breadboard
- Jumper wires (Male–Male or Male-Female)
- USB cable for Arduino

Procedure:

1. Connect HC-SR04 sensor (Echo pin to D12, Trigger pin to D13, VCC to 5V, GND to GND) on a breadboard or to the sensor directly using jumper wires.
2. Using Arduino IDE, upload the Arduino code to measure pulse width and calculate object distance using the pulseIn function.
3. Place obstacles in front of the sensor and observe the readings on the Serial Monitor.

Reference Circuit:

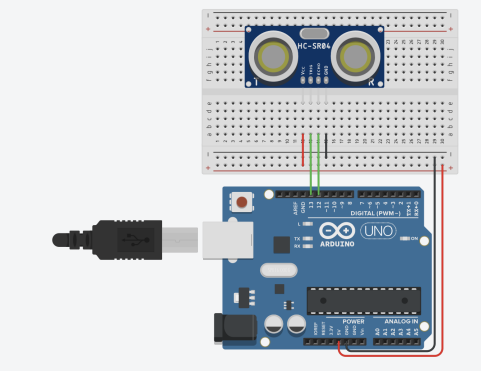

Figure 1: Circuit for HC-SR04 Ultrasonic sensor.

Physical Circuit:

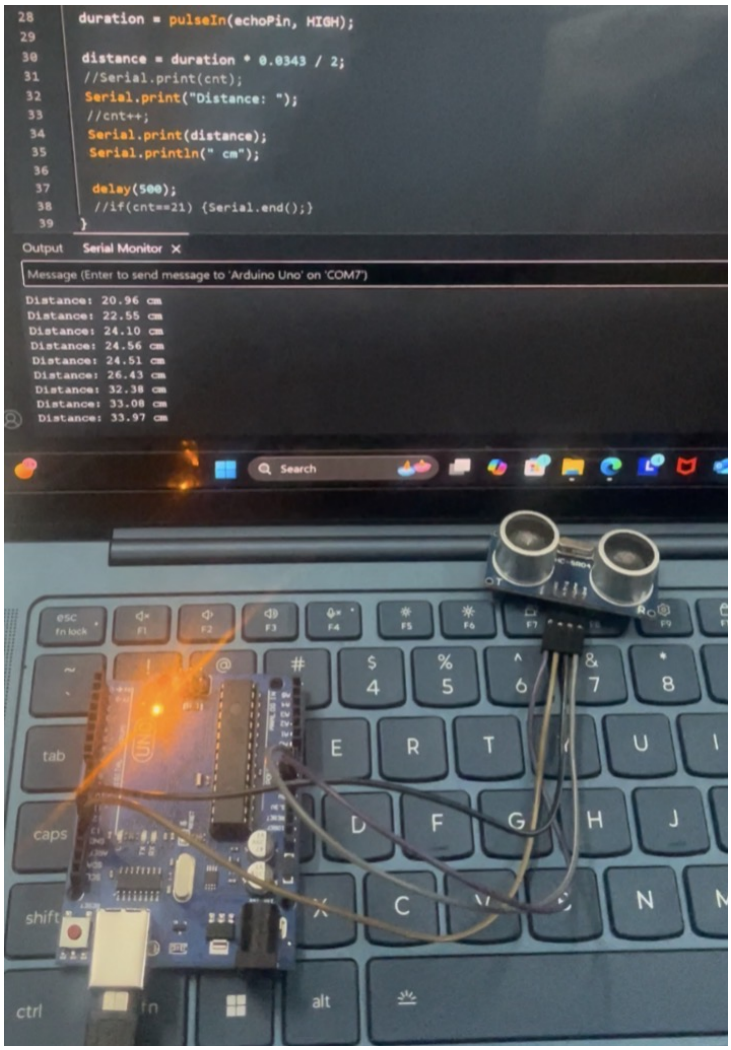

Figure 2: Working of HC-SR04 Ultrasonic sensor.

Code:

```
int trigPin = 13;
int echoPin = 12;
long duration;
float distance;

void clearSerialMonitor() {
  for (int i = 0; i < 50; i++) {
    Serial.println();
  }
}

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  clearSerialMonitor();
  Serial.println("Ultrasonic Distance Measurement");
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

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

  duration = pulseIn(echoPin, HIGH);

  distance = duration * 0.0343 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(500);
}
```

Listing 1: Distance Meter using HC-SR04 and Serial Monitor

Observations:

Serial Monitor Data

Ultrasonic Distance Measurement

1. Distance: 45.94 cm
2. Distance: 53.01 cm
3. Distance: 41.38 cm
4. Distance: 1192.99 cm
5. Distance: 7.58 cm
6. Distance: 9.12 cm
7. Distance: 8.21 cm
8. Distance: 17.34 cm
9. Distance: 22.11 cm
10. Distance: 1193.07 cm
11. Distance: 44.40 cm
12. Distance: 34.68 cm
13. Distance: 43.53 cm
14. Distance: 36.20 cm
15. Distance: 49.56 cm
16. Distance: 45.70 cm
17. Distance: 44.49 cm
18. Distance: 22.42 cm
19. Distance: 18.93 cm
20. Distance: 6.60 cm

- The Serial Monitor updated distance readings as obstacles varied in position.
- Consistent measurements were observed for most objects within the range of 2 cm to 400 cm.
- Small, irregular, or angled surfaces produced less stable readings due to wave reflections.

Explanation:

- The HC-SR04 generates an ultrasonic pulse and measures round-trip time for distance calculation, directly demonstrating sensor interfacing fundamentals.
- The pulseIn function measures the duration of the echo signal, which is directly proportional to object distance.
- The distance is calculated as:

$$\text{Distance} = \frac{\text{pulse duration} \times \text{speed of sound}}{2}$$

Conclusion:

- Arduino UNO successfully measured distances and dynamically displayed results, confirming correct hardware-software integration.

References:

- Lab manual
- Arduino documentation

PART B:

Aim:

To interface a 16x2 LCD display to Arduino and observe display of custom messages and sensor data.

Components Required:

- Arduino UNO board
- 16x2 LCD display module
- 10kΩ potentiometer (for contrast control)
- Breadboard
- Jumper wires (Male–Male)

Procedure:

1. Connect LCD display (RS=12, E=11, D4=5, D5=4, D6=3, D7=2) with potentiometer for contrast.
2. Using Arduino IDE, upload and modify Arduino LiquidCrystal library code for custom messages.
3. Adjust potentiometer for clear display; test custom strings.

Reference Circuit:

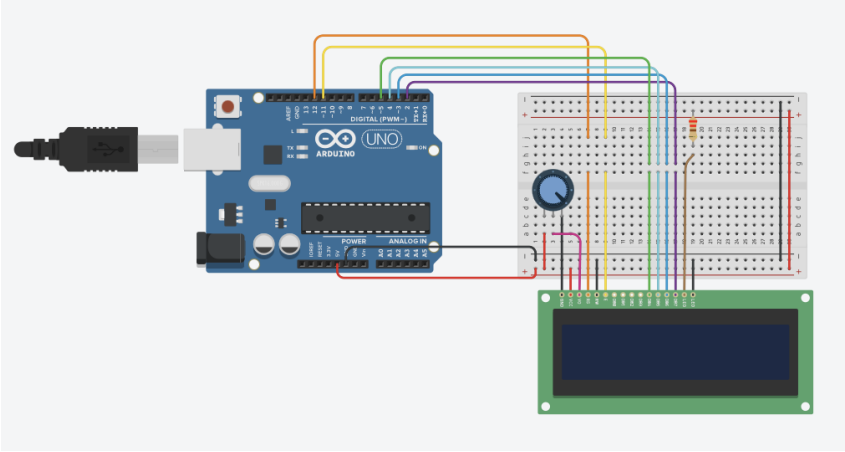

Figure 3: Circuit for 16x2 LCD module.

Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print(" hello world!");
}

void loop() {
}
```

Listing 2: LCD Display

Physical Circuit:

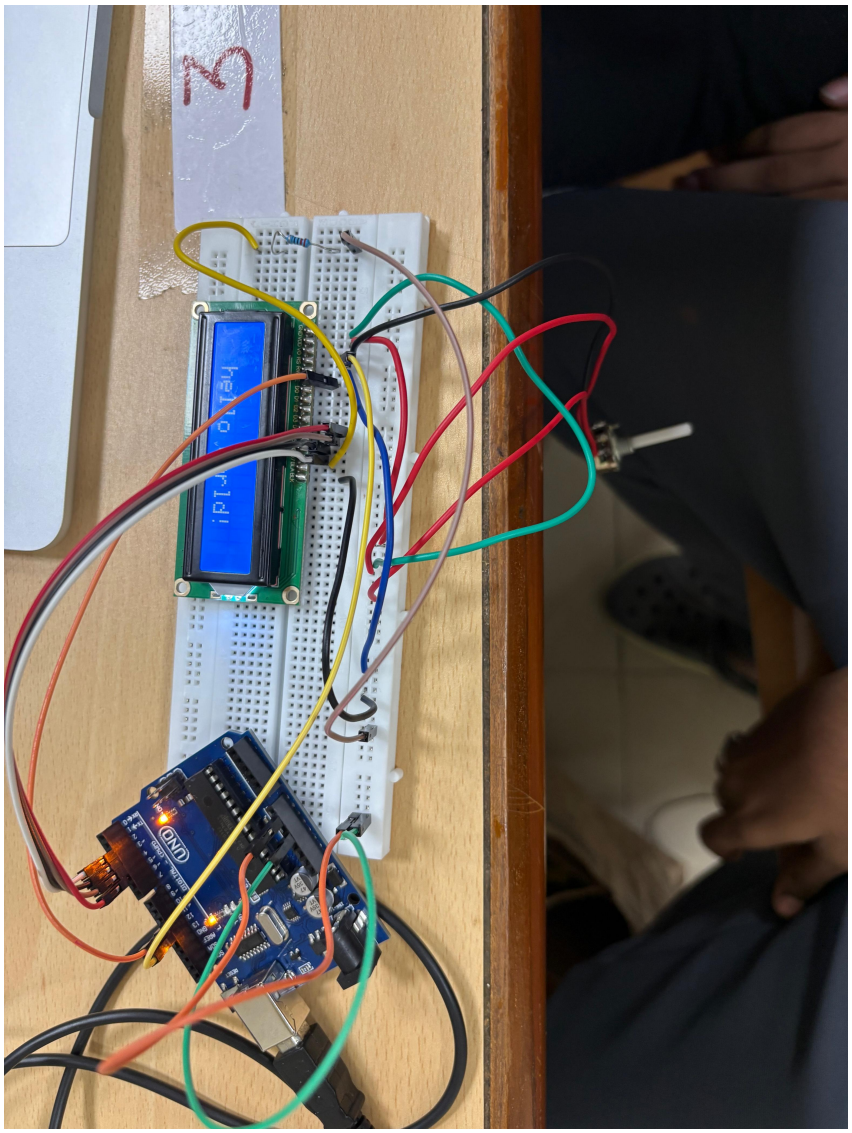

Figure 4: Working of 16x2 LCD module.

Observations:

- LCD displayed both default and custom messages after adjustment.
- Sensor data (distance) could also be routed to the LCD for real-time display.

Analysis:

- LCD initialization (‘lcd.begin(16,2)‘) and pin matching are essential for proper function.
- The LiquidCrystal library’s ‘lcd.print‘ and cursor positioning enable flexible display tasks.
- Integration of sensor and display demonstrates basic embedded system capabilities.

Conclusion:

- The LCD display project validated Arduino’s ability to drive visual output for project use.

PART C:

Aim:

To integrate the HC-SR04 Ultrasonic Sensor with a 16x2 LCD and display the measured distance in real time on the LCD using Arduino.

Components Required:

- Arduino UNO board
- HC-SR04 Ultrasonic sensor
- 16x2 LCD display module
- 10kΩ potentiometer
- Breadboard
- Jumper wires (Male–Male)

Procedure:

1. Connect both the ultrasonic sensor and LCD to the Arduino as described in Parts A and B.
2. Include the required libraries (‘LiquidCrystal.h‘) at the beginning of the Arduino code.
3. Define all sensor and LCD pin assignments in the code.
4. In the setup function, initialize both the LCD and trigger/echo pins.
5. In the loop, measure the distance using the ultrasonic sensor.
6. Use ‘lcd.clear()‘, ‘lcd.setCursor()‘, and ‘lcd.print()‘ to display distance readings on the LCD display.

Reference Circuit:

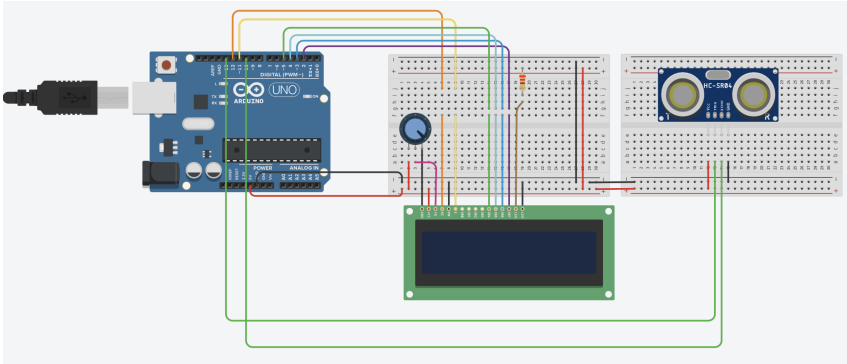

Figure 5: Wiring and integration of HC-SR04 sensor and 16x2 LCD with Arduino.

Physical Circuit:

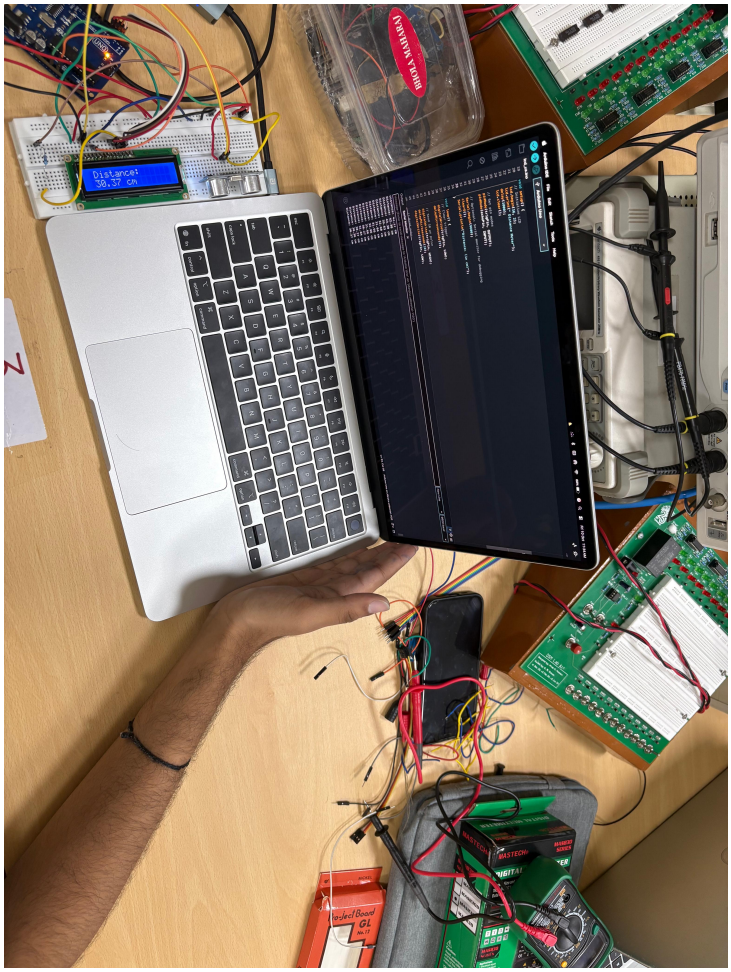

Figure 6: Working of HC-SR04 sensor and 16x2 LCD with Arduino.

Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int trigPin = 13;
const int echoPin = 10;

long duration;
float distance;

void clearS() {
  for (int i = 0; i < 50; i++) {
    Serial.println();
  }
}

void setup() {
  lcd.begin(16, 2);
  lcd.print(" Distance Meter");
  delay(1000);
  lcd.clear();

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);
  clearS();
  Serial.println("Distance: (in cm)");
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

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

  duration = pulseIn(echoPin, HIGH);

  distance = duration * 0.0343 / 2;

  lcd.setCursor(0, 0);
  lcd.print(" Distance:");
  lcd.setCursor(0, 1);
  lcd.print(" ");
  lcd.print(distance);
  lcd.print(" cm   ");

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(50);
}
```

Listing 3: Distance Meter using HC-SR04 and LCD

Conclusion:

- Integration of ultrasonic sensor and LCD with Arduino was achieved, with real-time distance display validating effective hardware and software interface.

References:

- Lab manual
- Arduino documentation