**Measurement using Ultrasonic Sensor**

**📏 Distance Measurement Using Ultrasonic Sensor and Arduino with I2C LCD**

**📘 1. Project Overview**

This project uses an **HC-SR04 ultrasonic sensor** connected to an **Arduino Uno** to measure the **distance to an object** in centimeters. The result is displayed in real time on a **16x2 LCD with I2C module**.

**🎯 2. Objectives**

* Measure distance using the HC-SR04 ultrasonic sensor.
* Display the measured distance on a 16x2 LCD via I2C.
* Create a simple and effective proximity/distance sensing system.

**🧰 3. Components Required**

| **Component** | **Quantity** |
| --- | --- |
| Arduino Uno | 1 |
| HC-SR04 Ultrasonic Sensor | 1 |
| 16x2 LCD with I2C Module | 1 |
| Breadboard (optional) | 1 |
| Jumper Wires | As needed |
| USB Cable for Arduino | 1 |

**🔌 4. Circuit Diagram / Connections**

**(URL:-** [**https://app.cirkitdesigner.com/project/d73786ca-0fba-40fe-ade3-a95035edc00f**](https://app.cirkitdesigner.com/project/d73786ca-0fba-40fe-ade3-a95035edc00f)**)**

**🟦 HC-SR04 Ultrasonic Sensor to Arduino**

| **HC-SR04 Pin** | **Arduino Pin** |
| --- | --- |
| VCC | 5V |
| GND | GND |
| TRIG | D9 |
| ECHO | D10 |

**🟩 I2C LCD to Arduino**

| **LCD Pin** | **Arduino Pin** |
| --- | --- |
| VCC | 5V |
| GND | GND |
| SDA | A4 |
| SCL | A5 |

**Note**: Adjust the LCD I2C address (0x27) if necessary using an I2C scanner.

**🧾 Libraries You Need**

* LiquidCrystal\_I2C

***Install them via the Arduino Library Manager (Sketch → Include Library → Manage Libraries).***

**💻 5. Arduino Code**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

// Initialize LCD: 0x27 is a common I2C address for 16x2 LCDs

LiquidCrystal\_I2C lcd(0x27, 16, 2);

const int trigPin = 9;

const int echoPin = 10;

long duration;

float distance;

void setup() {

Serial.begin(9600);

lcd.init(); // Initialize the LCD

lcd.backlight(); // Turn on the backlight

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

// Send ultrasonic pulse

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Read echo pulse

duration = pulseIn(echoPin, HIGH, 30000); // Timeout after 30ms

distance = duration \* 0.034 / 2; // Convert to cm

// Show on Serial Monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Show on LCD

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Distance:");

lcd.setCursor(0, 1);

if (duration == 0) {

lcd.print("Out of Range");

} else {

lcd.print(distance, 2);

lcd.print(" cm");

}

delay(500);

}

**📏 6. Working Principle**

* The **HC-SR04** emits an ultrasonic sound wave via the **TRIG** pin.
* It listens for the **echo** through the **ECHO** pin.
* The time taken for the echo to return is used to calculate distance:

Distance (cm)=Time (μs)×0.0342\text{Distance (cm)} = \frac{\text{Time (μs)} \times 0.034}{2}

* The result is displayed on both the **Serial Monitor** and the **I2C LCD**.

**📋 7. Applications**

* Proximity sensor in robotics.
* Object detection systems.
* Distance-based automation (e.g., automatic doors, obstacle avoiders).
* Educational demo for wave propagation and sensing.

**⚠️ 8. Limitations**

* Effective range: **2 cm to ~400 cm**.
* Requires **a flat surface** for accurate reflection.
* Not suitable for soft or irregular objects (sound absorption/scattering).