MCI Project on Topic

Ultrasonic Distance Measurement

For

2EC701CC23
Microcontroller and Interfacing

B. Tech. Semester IV

Submitted to: Dr. Rutul Patel

Submitted By:

Divyanshu Kalal
Roll No: 23BEC053

&

Dharm Dave
Roll No: 23BEC044

Institute of Technology
Nirma University
Ahmedabad, Gujarat, 382481

April 6, 2025

Table of Contents

Contents

Abstract

Accurate distance measurement plays a vital role in robotics, automation, and obstacle detection. This project presents a real-time distance sensing system using an **HC-SR04 ultrasonic sensor** interfaced with an **AT89C51 microcontroller**, with output displayed on a **16x2 LCD**.

The system functions by transmitting ultrasonic pulses and measuring the echo return time to calculate distance. The microcontroller processes this data using its internal timer and displays the result in centimeters.

This work highlights key aspects of *8051 programming*, including *embedded C*, *sensor interfacing*, and *timing-based computations*. It provides a practical foundation for further development in *automation* and *IoT-integrated smart systems*.

1 Introduction

1.1 Need for Distance Measurement

Distance measurement is vital in **automation, robotics, industrial monitoring**, and **vehicle safety**. It enables **collision avoidance, object tracking**, and **level sensing**.

Ultrasonic sensors are preferred for their **non-contact operation, durability**, and **reliable performance**, making them ideal for real-time applications like autonomous navigation and smart infrastructure.

1.2 Microcontroller's Role

Microcontrollers act as the **brain** of embedded systems. They handle sensor data, process inputs, and control outputs.

Key tasks include:

- Reading sensor signals
- Processing data
- Displaying results
- Controlling actuators

Their compact size and flexibility suit automation systems perfectly.

1.3 Why AT89C51?

The **AT89C51** microcontroller is chosen for its **simplicity, affordability**, and features like:

- 8051-based architecture

- Sufficient I/O pins for sensor and LCD
- Built-in timers for precise timing
- 4 KB flash memory

These features make it ideal for small-scale, real-time projects like distance measurement.

2 Components and Specifications

This project employs several key electronic components for precise distance measurement and display. The following table summarizes each component along with its specifications and purpose.

Component	**Specification**	**Purpose**
AT89C51	8-bit MCU, 4 KB Flash	Central controller for signal processing
HC-SR04	2–400 cm Range, 40 kHz	Measures distance using ultrasound
16x2 LCD	ASCII Compatible, 5V	Displays the measured distance
Power Supply	5V DC Regulated	Provides stable power to components
Crystal Oscillator	11.0592 MHz	Generates clock signal for microcontroller
Capacitors	10 µF, 33 pF	Filtering and power stabilization
Resistors	1 kΩ, 10 kΩ	Signal control and pull-up functions
Connecting Wires	Standard jumpers	Wiring connections between modules

Table 1: List of Components and Specifications

2.1 AT89C51 Microcontroller

The **AT89C51** is an 8051-based 8-bit microcontroller with:

- **4 KB** Flash memory and **128 Bytes** RAM
- **32 I/O pins** divided into four 8-bit ports
- Integrated **timers/counters** and **serial communication**
- Operating voltage: $\mathbf{4V - 5.5V}$

It serves as the system's core, handling timing, processing, and display operations.

2.2 HC-SR04 Ultrasonic Sensor

This sensor performs non-contact distance measurement by transmitting ultrasonic pulses and timing the returning echo. Key features include:

- Voltage: **5V DC**
- Range: **2 cm to 400 cm**
- Accuracy: **±3 mm**
- Frequency: **40 kHz**
- Minimum Trigger Pulse: **10 µs**

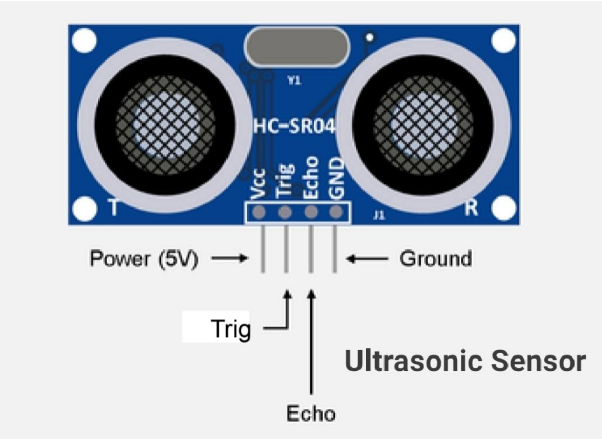

Figure 1: Pin Diagram of HC-SR04

2.3 16x2 LCD Display

A **16x2 alphanumeric LCD** is used to show real-time distance values. It supports:

- Voltage: **5V DC**
- **ASCII characters and custom symbols**
- Both **4-bit and 8-bit** communication modes

2.4 Power Supply (5V DC)

A regulated **5V DC** source, typically using a **7805 voltage regulator**, ensures a steady supply to all components.

2.5 Crystal Oscillator (11.0592 MHz)

The **11.0592 MHz** crystal provides accurate timing for the microcontroller and ensures precise serial communication.

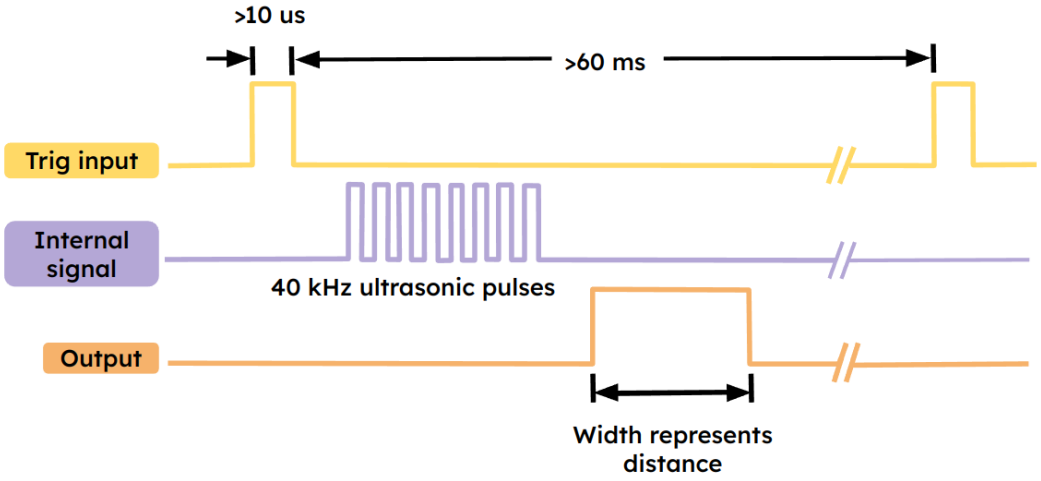

Figure 2: Pulse Waeforms

2.6 Passive Components

- **Capacitors** (10 µF, 33 pF) for power filtering and oscillator stability.
- **Resistors** (1 kΩ, 10 kΩ) for signal conditioning and pull-up configurations.
- **Jumper wires** for making connections on the breadboard or PCB.

3 Working Principle

The system measures distance using the **HC-SR04 ultrasonic sensor**, which operates on the principle of *echo reflection*. The AT89C51 microcontroller processes the signal and displays the result on a 16x2 LCD.

3.1 Operation Overview

1. The microcontroller sends a **10 µs trigger pulse** to the HC-SR04.
2. The sensor emits a **40 kHz ultrasonic wave**, which travels through air at **343 m/s**.
3. When the wave hits an object, it reflects back and is received by the sensor.
4. The sensor outputs a pulse whose width corresponds to the round-trip time.
5. The microcontroller measures this pulse width to calculate distance using:

$$\text{Distance (cm)} = \frac{\text{Pulse Width (µs)} \times 0.0343}{2} \tag{1}$$

3.2 Display and Update

The calculated distance is:

- Converted into a readable format,
- Sent to the LCD,
- Updated continuously to show real-time values.

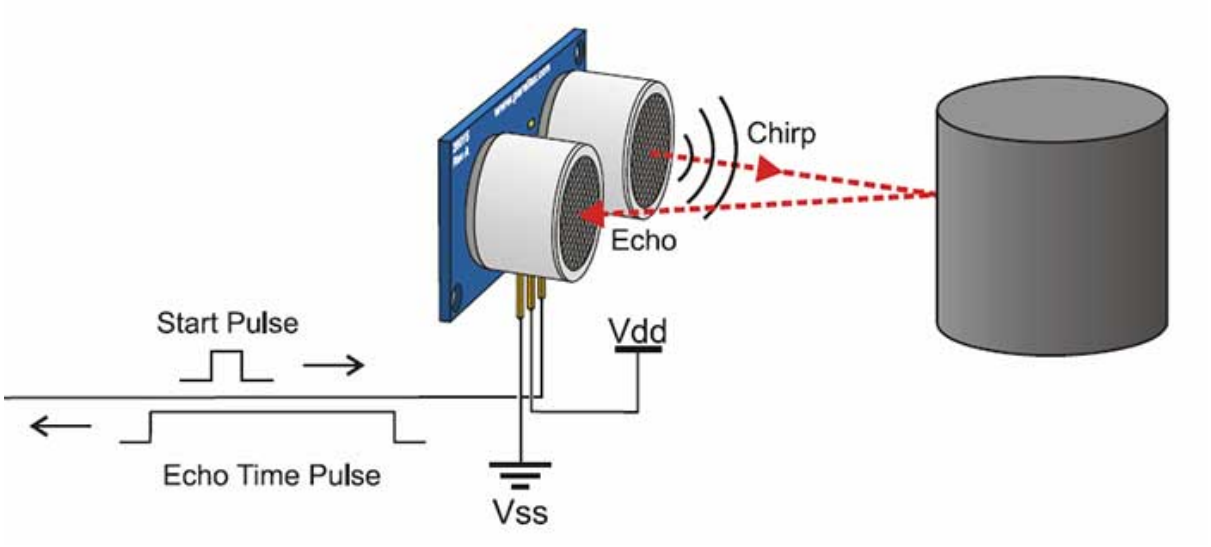

Figure 3: Working principle of the ultrasonic distance measurement system

3.3 Factors Affecting Accuracy

- **Temperature:** Affects speed of sound, impacting distance accuracy.
- **Surface type:** Soft surfaces may absorb waves, weakening the echo.
- **Range:** Optimal measurement between **2–400 cm**.
- **Alignment:** Sensor must face the object directly for best results.

4 Circuit Diagram

The circuit connects the **AT89C51 microcontroller** to an **HC-SR04 ultrasonic sensor** and a **16x2 LCD display** to measure and display distance efficiently.

4.1 Key Interconnections

- **HC-SR04 Sensor:**
  - VCC → +5V, GND → Ground

- Trigger → *P3.5*, Echo → *P3.2*

- **16x2 LCD Display:**
  - RS → *P3.1*, RW → *P3.3*, EN → *P3.4*
  - Data Pins D0–D7 → *P2.0 to P2.7*
  - VSS → GND, VDD → +5V, VEE → Contrast control (via 10kΩpot)

- **Clock Circuit:**
  - 11.0592 MHz crystal between XTAL1 and XTAL2
  - Two 33 pF capacitors to ground

- **Reset Circuit:**
  - 10 µF capacitor and 10kΩresistor on the RST pin

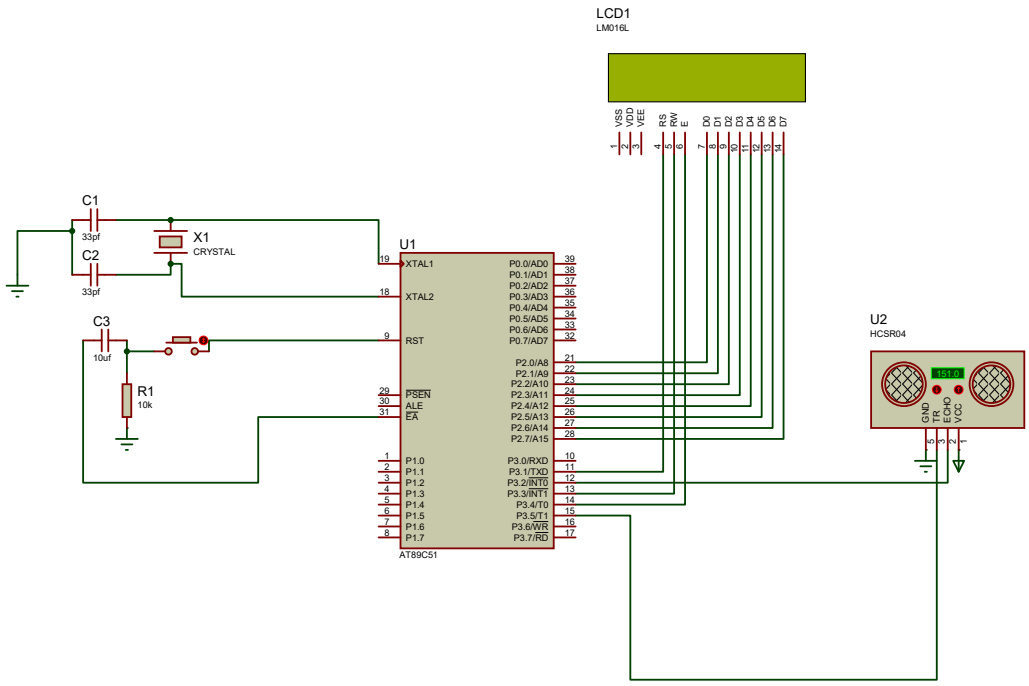

Figure 4: Circuit diagram of the ultrasonic distance measurement system

4.2 Working Summary

The microcontroller sends a trigger signal to the ultrasonic sensor, receives the echo, calculates the distance, and displays it on the LCD.

5 Algorithm & Flowchart

The system follows a looped process to continuously measure and display distance using the HC-SR04 sensor and AT89C51 microcontroller.

5.1 Algorithm

1. **Initialize** microcontroller ports and LCD.
2. Configure Trigger as output and Echo as input.
3. Send a 10 µs pulse to the Trigger pin.
4. Wait for Echo and measure its pulse width.
5. Calculate distance using:

$$\text{Distance (cm)} = \frac{\text{Time} \times 0.0343}{2} \tag{2}$$

6. Display distance on the LCD.
7. Repeat the process after a short delay.

5.2 Flowchart

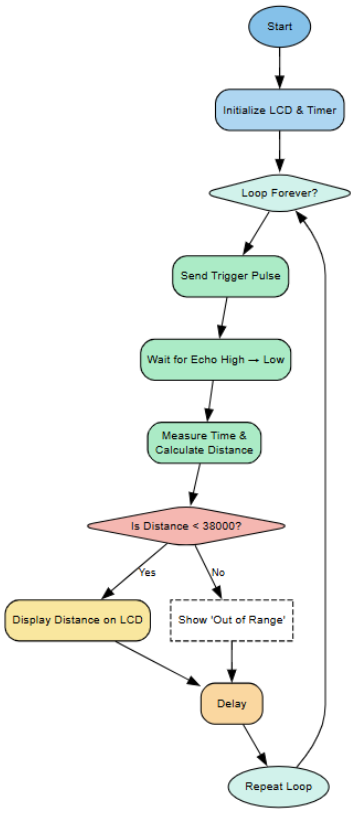

Figure 5: Flowchart of the Distance Measurement Process

5.3 Execution Cycle

- The loop runs continuously for real-time distance updates.
- A delay is added for stable readings.
- If no object is detected, the LCD shows *"Out of Range"*.

6 Results & Observations

The ultrasonic distance measurement system was successfully implemented using the **AT89C51 microcontroller** and **HC-SR04 sensor**. The system accurately measured distances and displayed them in real-time on a **16x2 LCD screen**.

6.1 Sample Output

Below are sample readings observed during testing:

Actual Distance (cm)	Measured Distance (cm)
10	10
20	19
30	30
50	49

6.2 Challenges and Solutions

- **Fluctuating Readings:** Initial measurements were inconsistent due to ambient noise. A basic *software averaging filter* was implemented to stabilize output.
- **LCD Malfunction:** Display glitches occurred due to timing issues. This was resolved by introducing appropriate *delays* between LCD commands.
- **Power Instability:** Occasional resets were caused by voltage drops. Adding a *decoupling capacitor* near the V_{CC} pin improved stability.

Project Demonstration Video

`Click here to view`

7 Conclusion

The ultrasonic distance measurement system built using the **AT89C51 microcontroller** and **HC-SR04 sensor** successfully measured distances between **2 cm and 400 cm** with reasonable accuracy. The use of a **16x2 LCD** enabled clear, real-time distance display, making the setup both efficient and easy to use.

7.1 Future Scope

This system can be further improved through:

- **IoT Integration**: Connect to the cloud using Wi-Fi or NB-IoT for remote monitoring.
- **Wireless Communication**: Transmit data via Bluetooth or RF modules instead of using a local display.
- **Obstacle Detection**: Use multiple sensors for collision avoidance in robotics.
- **Advanced Displays**: Replace the LCD with a GLCD or OLED for better data visualization.

8 References

The following references were used during the design and implementation of the project:

1. *HC-SR04 Ultrasonic Sensor Datasheet*. Available at: `https://www.electronicsdatasheets.com`
2. *AT89C51 Microcontroller Datasheet*. Available at: `https://www.atmel.com/microcontroller`
3. Mazidi, M. A., *Embedded C Programming for 8051 Microcontrollers*, Pearson Education.
4. *LCD Interfacing with 8051 Microcontroller*. Available at: `https://www.embedded.com`
5. *"Ultrasonic Distance Measurement using Microcontrollers"*, IEEE Xplore. Available at: `https://ieeexplore.ieee.org`

A Appendix

The following is the Embedded C code for the ultrasonic distance measurement system:

C CODE FILE

```
// Include required header files
#include <reg51.h>       // AT89C51 microcontroller definitions
#include "lcd.h"         // LCD function library

// Define Trigger and Echo pins
#define TRIG P2_0
#define ECHO P2_1

// Function declarations
void delay_ms(unsigned int);
void trigger_ultrasonic();
unsigned int measure_echo_time();
void display_distance(unsigned int);

void main() {
    unsigned int distance;

    lcd_init();                  // Initialize LCD
    lcd_clear();
    lcd_print("Ultrasonic Sensor");

    while (1) {
        trigger_ultrasonic();                    // Send trigger
            signal
        distance = measure_echo_time() / 58;     // Convert to
           cm
        display_distance(distance);              // Show on LCD
        delay_ms(500);                           // Delay
           between measurements
    }
}

// Generate 10us trigger pulse
void trigger_ultrasonic() {
    TRIG = 1;
    delay_ms(10);
    TRIG = 0;
}

// Measure the duration of echo pulse
unsigned int measure_echo_time() {
    unsigned int time = 0;
    while (!ECHO);               // Wait for echo HIGH
```

```
    while (ECHO) time++;          // Count until echo LOW
    return time;
}

// Display the measured distance on LCD
void display_distance(unsigned int distance) {
    lcd_clear();
    lcd_print("Distance: ");
    lcd_print_int(distance);      // Convert int to string
    lcd_print(" cm");
}

// Millisecond delay using nested loops
void delay_ms(unsigned int ms) {
    unsigned int i, j;
    for (i = 0; i < ms; i++)
        for (j = 0; j < 1275; j++);
}
```