RAMAIAH SKILL ACADEMY

Project Report

on

Touch Activated Distance Measurement

ELECTRONICS AND COMMUNICATION ENGINEERING

EMBEDDED SYSTEM DESIGN

Submitted by

AVINASHA J K

RSAGWPESD03

Abstract

This project focuses on designing a system for touch-activated distance measurement by interfacing a touch sensor and an ultrasonic distance sensor with a microcontroller at the register level. The system is activated when the user touches the sensor, triggering the ultrasonic sensor to measure the distance to an object. The measured distance is then displayed on an output device, such as an LCD or a seven-segment display. The use of register-level programming ensures efficient and precise control of the microcontroller by directly configuring and manipulating hardware registers without relying on high-level libraries.

This approach enhances the understanding of low-level hardware-software interactions and enables better optimization of the system. The project highlights the integration of sensor inputs and real-time data processing to provide a user-activated measurement system. This system has potential applications in industrial automation, robotics, and assistive technologies, where precise and interactive measurement systems are required. It also serves as a learning platform for embedded systems enthusiasts, offering insights into the use of touch sensors, ultrasonic sensors, and microcontroller programming at a fundamental level.

Table of Contents

Chapter 1

INTRODUCTION

In various fields such as automation, robotics, and industrial applications, distance measurement is an essential function. By minimising useless tasks, combining this function with touch-based activation optimises user interaction and system efficiency. This project displays a touch-activated distance measuring device that combines an ultrasonic sensor, a touch sensor, and a microcontroller with register-level programming. When a touch input is detected, the system turns on, uses an ultrasonic sensor to calculate distance, and shows the result on an output device, either an LCD or a seven-segment display.

The project focusses direct control over the microcontroller's hardware resources by avoiding high-level abstractions through the use of register-level programming. Accurate timing, effective use of resources, and a better comprehension of the underlying hardware operations are made possible by this method. The project serves as a basis for developing more complex, real-time interactive systems in addition to showcasing the useful application of sensor interfacing and low-level programming. Applications needing on-demand distance measurement can be served by the system's touch-activated feature, which also makes it user-friendly and energy-efficient.

Chapter 2

Scope and Objective

❖ Objective

This project's main goal is to use register-level programming to create a touch-activated distance measurement device. This comprises:

Efficient Sensor Integration: Touch-based activation and accurate distance measurement are made possible by connecting an ultrasonic and touch sensor to a microprocessor.

Low-level programming mastery: Implementing the system at the register level for achieving accurate control, effective resource usage, and improved performance is known as low-level programming mastery.

Real-Time Measurement: Making certain that, when activated, the system gives precise and instantaneous,distance,readings.

User-Friendly Interaction: Developing an interactive system that only activates in response to user input, hence maximising usability and minimising energy use.

Application Demonstration: Highlighting the possibilities of touch-activated devices in real-world settings such assistive technology, automation, and robotics.

❖ Scope

- **Educational Scope**:
  - Provides a hands-on learning experience in low-level microcontroller programming and sensor interfacing.
  - Demonstrates the use of touch sensors and ultrasonic sensors in real-time embedded systems.
- **Research and Development**:
  - Serves as a foundation for designing energy-efficient, touch-based control systems.
  - Can be extended to more complex applications, such as obstacle detection in robotics or automated systems requiring user interaction.
- **Practical Applications**:
  - Ideal for industrial automation systems requiring precise, on-demand distance measurement.
  - Useful in robotics for proximity detection and navigation.
  - Applicable in assistive technologies, such as touch-activated tools for individuals with limited mobility.

Chapter 3

METHODOLOGY AND IMPLEMENTATION

3.1 Methodology

Block Diagram

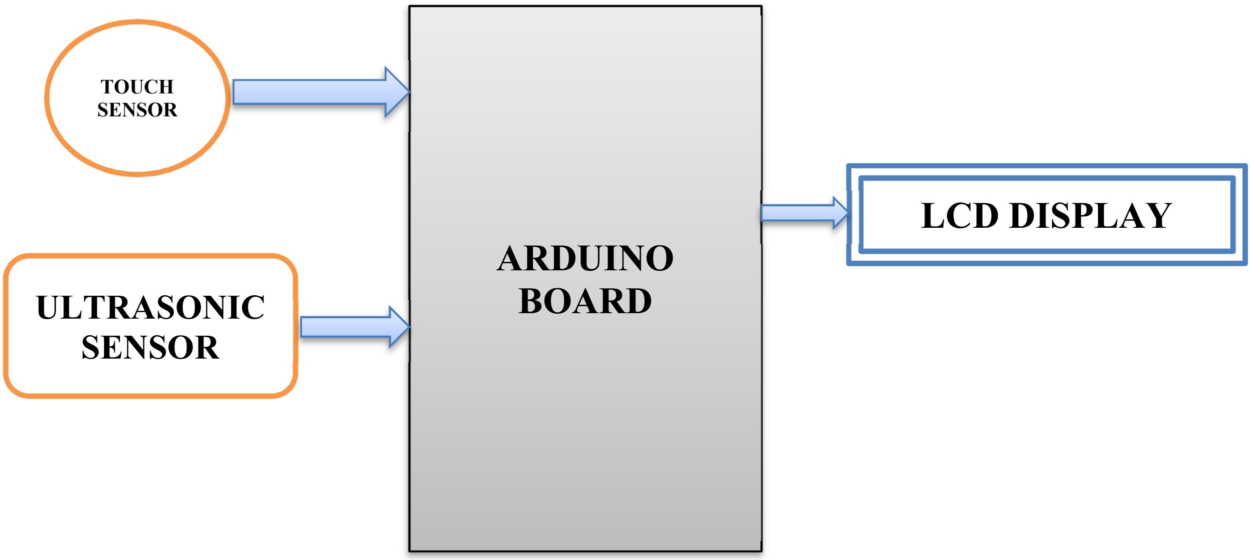


Fig.1 Block Diagram of Touch Activated Distance Measurement

Working: This diagram illustrates an Arduino-based system involving a touch sensor, an ultrasonic sensor, and an LCD display. The core component is the Arduino board, which interfaces with the sensors to process data and display results. The touch sensor and ultrasonic sensor provide input to the Arduino, which then processes this data and outputs it to the LCD display, showing the distance measured by the ultrasonic sensor in centimeters.

Arduino Sensor System SOP
1. Connect the Touch Sensor to the Arduino Board.
2. Connect the Ultrasonic Sensor to the Arduino Board.
3. Program the Arduino to read input from the Touch Sensor.
4. Configure the Ultrasonic Sensor to measure distance and send data to the Arduino.
5. Ensure the Arduino processes the distance data to output in centimeters.
6. Connect the LCD Display to the Arduino Board.
7. Display the processed distance data on the LCD Display.
8. Verify that the system operates correctly by checking the LCD Display for accurate distance readings.

Circuit Diagram

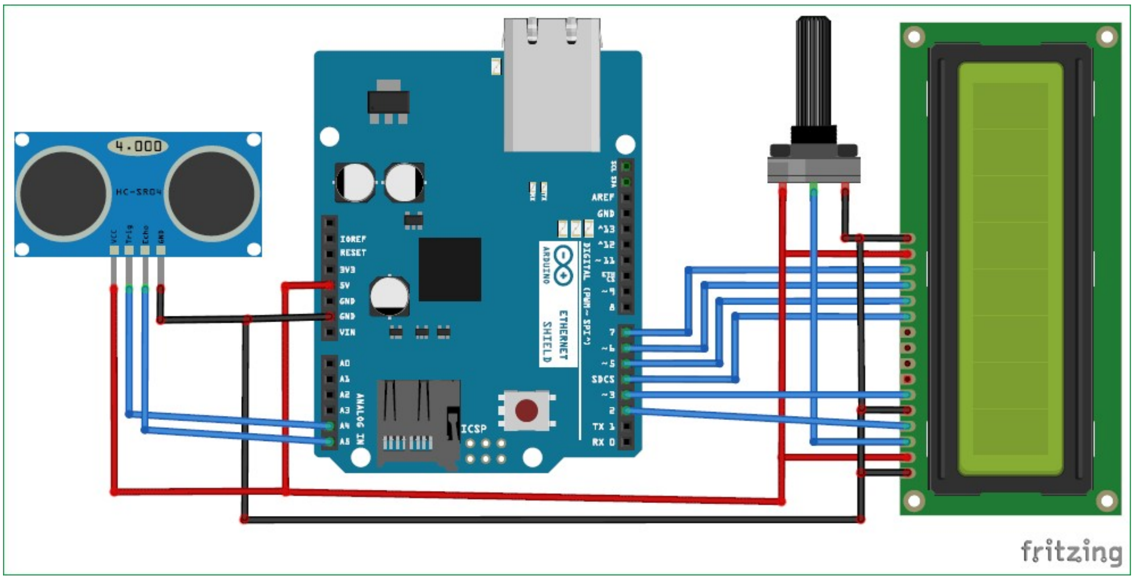


Fig.2 Circuit Diagram of Touch Activated Distance Measurement

The circuit diagram for **arduino and ultrasonic sensor** is shown above to measure the distance whenever the Touch is activated.In circuit connections Ultrasonic sensor module's "trigger" and "echo" pins are directly connected to pin 18(A4) and 19(A5) of arduino. A 16x2 LCD is connected with arduino in 4-bit mode. Control pin RS, RW and En are directly connected to arduino pin 2, GND and 3. And data pin D4-D7 is connected to 4, 5, 6 and 7 of arduino.

First of all we need to trigger the ultrasonic sensor module to transmit signal by using Arduino and then wait for receive ECHO. Arduino reads the time between triggering and Received ECHO. We know that speed of sound is around 340m/s. so we can calculate distance by using given formula:

Distance= (travel time/2) * speed of sound.

Where speed of sound around 340m per second.

A 16x2 LCD is used for displaying distance.

3.2 Hardware Requirements

- **Arduino Uno:** Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a micro controller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.The Arduino platform has become quite popular with people just starting out with electronics, and for good reason.Unlike most previous programmable circuit boards, the Arduino does not needa separate piece of hardware (called a programmer) in order to load new code onto the board you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible pack.The Uno is one of the more popular boards in the Arduino family and a great choice for beginners. We'll talk about what's on it and what it can do later in the talk about what's on it and what it can do later in the tutorial. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming.

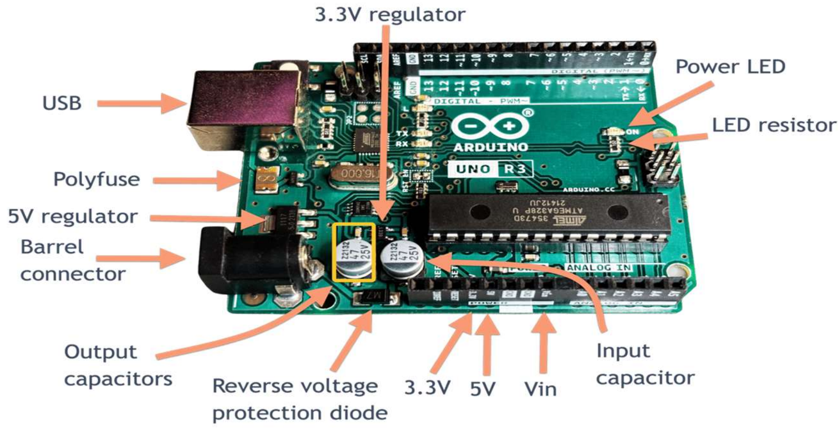


Fig.3 Arduino Uno

- **LCD:** An **LCD (Liquid Crystal Display)** is used in this project to display the measured distance, providing a clear visual output. It typically operates with a microcontroller via GPIO pins, requiring control signals like RS (Register Select), RW (Read/Write), and E (Enable) along with data lines. For this project, a 16x2 LCD is commonly used, capable of displaying two lines with 16 characters each. The LCD is interfaced using register-level programming to send commands (e.g., initialization, cursor position) and data (distance values). Proper timing delays and configuration are critical for its seamless operation.

Fig.4 LCD

- **Ultrasonic Sensor:** An ultrasonic Sensor is a device used to measure the distance between the sensor and an object without physical contact. This device works based on time-to-distance conversion. The HC-SR04 Ultrasonic Distance Sensor is an inexpensive device that is very useful for robotics and test equipment projects.The HC-SR04 can be hooked directly to an Arduino or other microcontroller and it operates on 5 volts. This ultrasonic distance sensor is capable of measuring distances between 2 cm to 400 cm.

  Working: Ultrasonic sensors measure distance by sending and receiving the ultrasonic wave. The ultrasonic sensor has a sender to emit the ultrasonic waves and a receiver to receive the ultrasonic waves. The transmitted ultrasonic wave travels through the air and is reflected by hitting the Object. Arduino calculates the time taken by the ultrasonic pulse wave to reach the receiver from the sender.

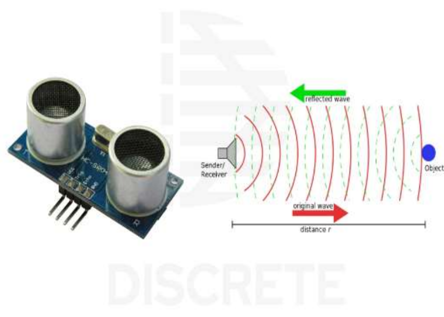


Fig.5 Ultrasonic Sensor

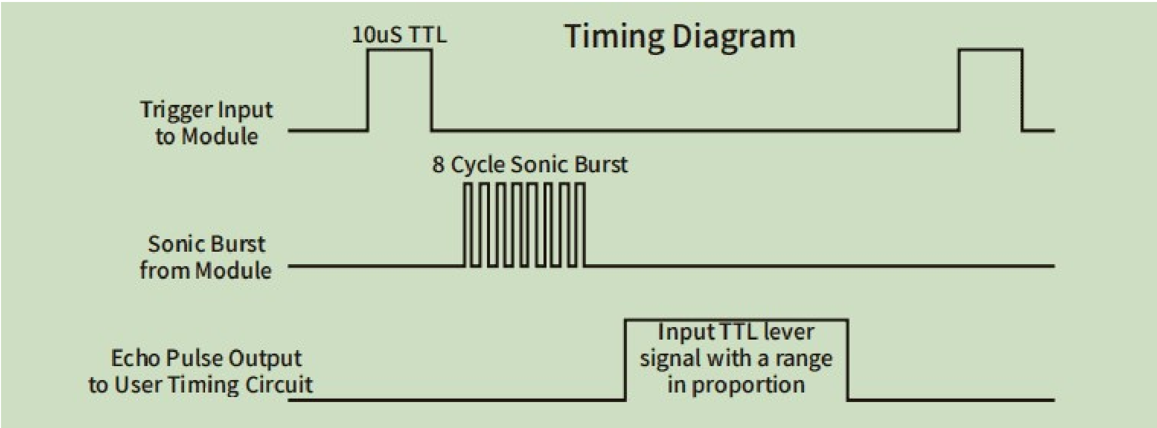


Fig.6 Timing Diagram of Ultrasonic Sensor

- **Touch Sensor:** The sensor consists of the TTP223 IC which is based on the capacitive sensing principle. It consists of a sensing electrode and an oscillator circuit.When a conductive material that is finger comes in contact with the sensing electrode it changes the capacitance of the electrode.This change is detected by the oscillator , circuit, then the oscillator circuit generates a digital output signal.This indicates the touch or proximity of the object. It is widely used in various electronic devices such as toys and touch switches.

Specifications:

The power supply for the sensor is 2 to 5.5V DC

The power supply for the sensor is 2 to 5.5V DC

Output high VOH is 0.8V VCC

Output low VOL is 0.3V VCC

Fig.7 Touch Sensor

- **Bread Board:** A breadboard was utilized in this project as a versatile and reusable platform for building the circuit without soldering. It served as an essential tool for testing and prototyping the connections between the touch sensor, LED, and microcontroller. The breadboard allowed easy placement and modification of components, ensuring a quick setup and seamless debugging process.

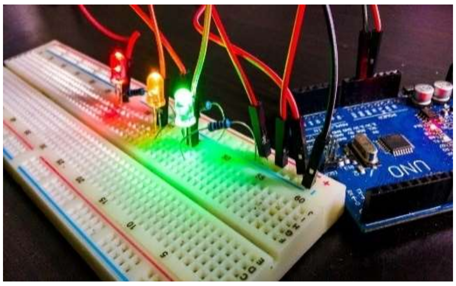


Fig.8 Bread Board

- **Resistor:** Resistors play a crucial role in ensuring proper functioning and protection of the electronic components. A resistor is used as a current-limiting device to safeguard the LED from excessive current, which could otherwise damage it. By limiting the current to a safe level, the resistor ensures the LED operates within its specified range, maintaining optimal brightness and longevity.

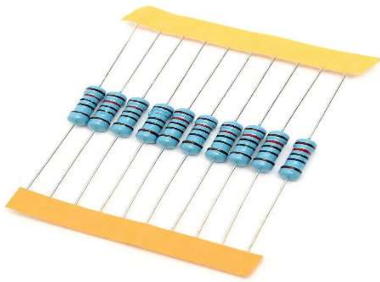


Fig.9 Resistor

- **Jumper wires:** Jumper wires are a vital component in this project, used to create temporary, flexible connections between the microcontroller, touch sensor, LED, and other components on a breadboard. These insulated wires are available in three types—male-to-male, male-to-female, and female-to-female—allowing seamless connection between headers, pins, and bread board.

Fig.10 Jumper Wires

3.3 Software Requirements

- **Arduino Integrated Development Environment(Arduino IDE):** is an open-source software tool used for writing, compiling, and uploading code to Arduino microcontroller boards. Designed to be simple and user-friendly, the IDE allows beginners and advanced users to develop embedded systems effortlessly. It features a text editor with syntax highlighting, automatic indentation, and error detection, making coding more efficient. Programs, known as "sketches," are written in a C/C++-based language and saved with a `no` extension. The IDE includes a built-in compiler that converts the written code into machine instructions, which are uploaded to the board using a USB connection. A notable feature is the **Serial Monitor**, which facilitates real-time communication and debugging between the computer and the Arduino board. The IDE supports various boards like Arduino UNO, Mega, Nano, and even third-party boards such as ESP8266 and ESP32, thanks to its **Board Manager**. It also allows easy integration of libraries, enabling users to include pre-written code for sensors, motors, and communication protocols like I2C, SPI, and UART. The IDE is cross-platform, running on Windows, macOS, and Linux, The IDE's simplicity, combined with extensive community support and open-source nature, makes it a powerful tool for learning and developing microcontroller-based projects. Users can download the software for free from the official Arduino website and begin coding by connecting their Arduino board, selecting the appropriate board and port, and uploading their sketch with ease. Overall, the Arduino IDE is an essential tool for prototyping and building innovative embedded systems.

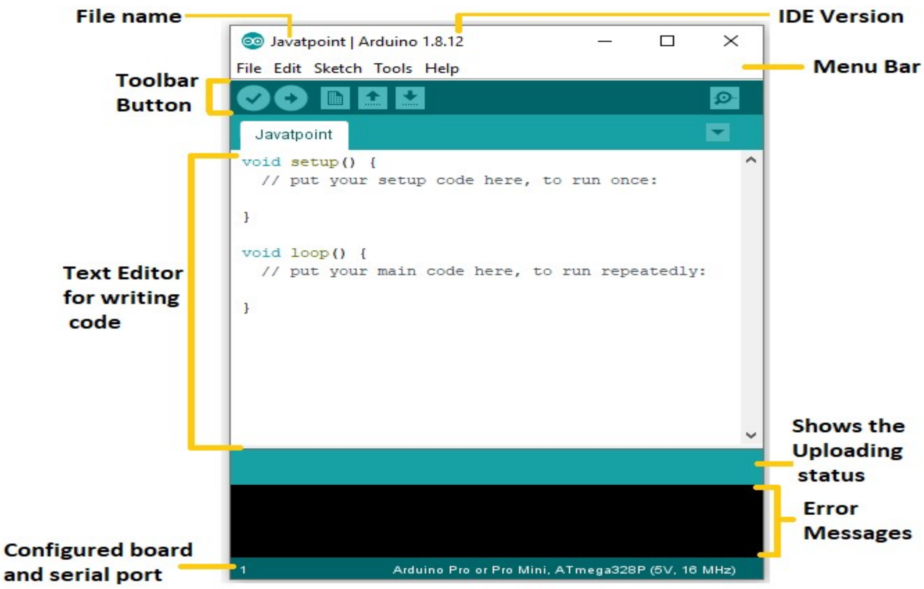


Fig.11 Arduino Ide

Register-Level Programming

3.1.1 Refers to programming microcontrollers or processors by directly accessing and manipulating their internal registers. Registers are small, fast storage locations within a microcontroller's CPU that control the operation of hardware peripherals like GPIO (General Purpose Input/Output), timers, ADC (Analog-to-Digital Converters), communication modules (SPI, I2C, UART), and other components. This approach allows developers to configure and control hardware with a high degree of precision and efficiency.

3.1.2 At the register level, programmers write code by understanding the memory map of the microcontroller and the specific bits within each register. Each register is typically mapped to a specific memory address, and each bit in the register corresponds to a specific function or hardware setting. For example, in a GPIO register, certain bits control

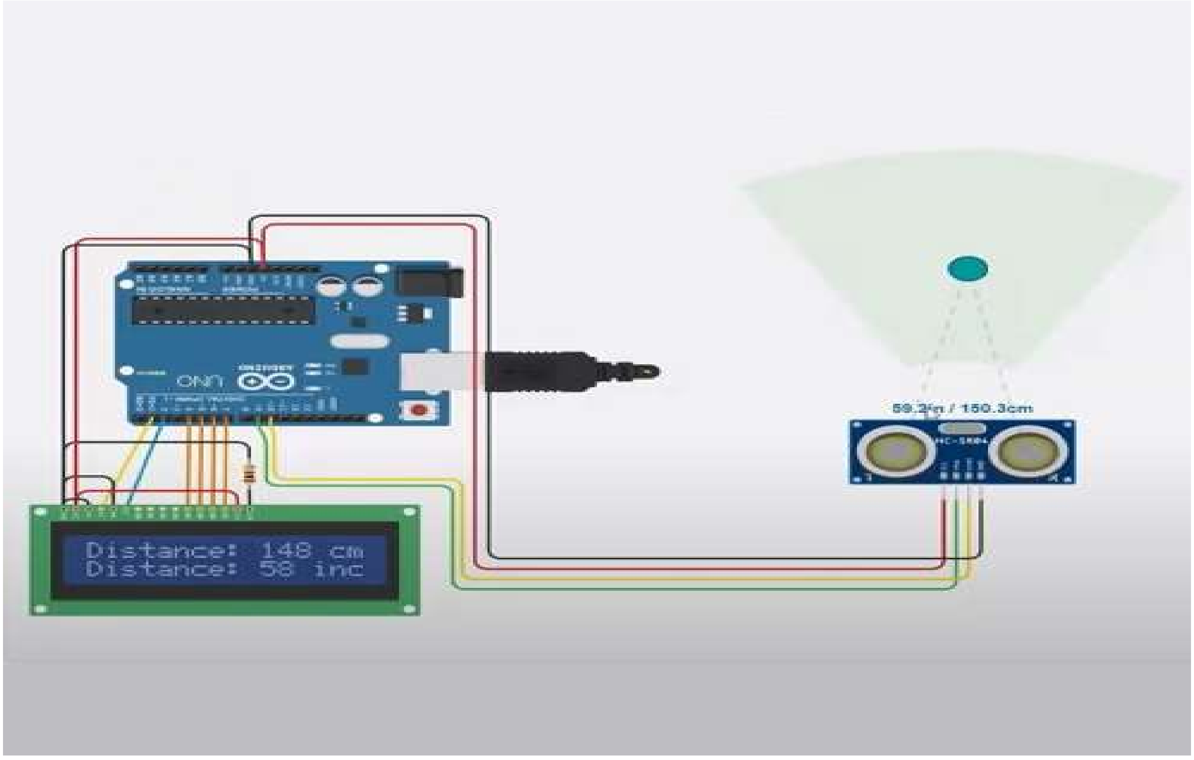
whether pins are set as input or output, while others determine their high or low states

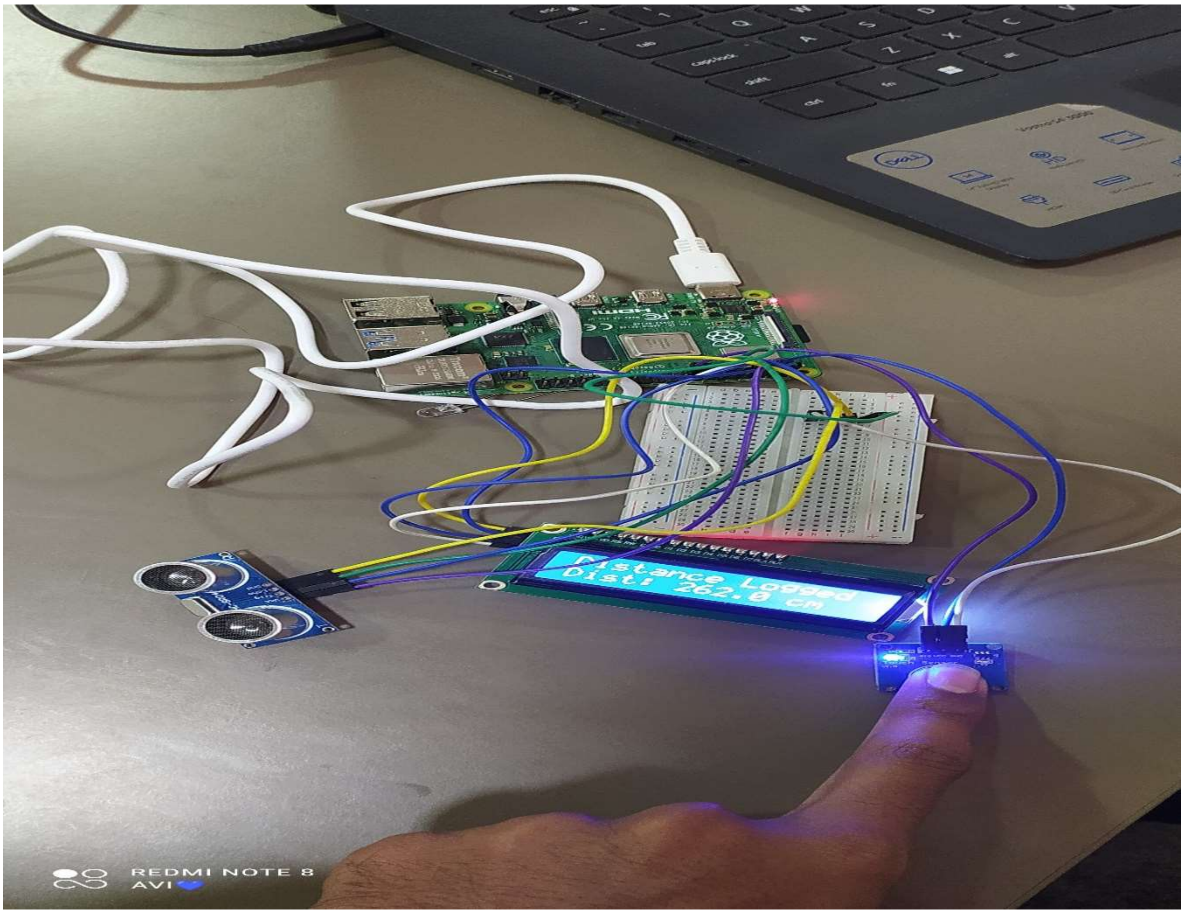
Algorithm

1. **System Initialization**
    - Initialize the microcontroller by configuring the necessary registers for GPIO, timer, and peripheral interfaces.
    - Set up the touch sensor as an input pin and configure the ultrasonic sensor (trigger pin as output and echo pin as input).
    - Initialize the output display (e.g., LCD or seven-segment display) to show measured distance.
2. **Idle State Monitoring**
    - Continuously monitor the touch sensor's status by reading the appropriate GPIO register.
3. **Touch Activation**
    - If a touch input is detected (logical HIGH or LOW depending on the touch sensor type):
        - Activate the ultrasonic sensor by sending a trigger pulse (configure a timer or delay for precise timing).
4. **Distance Measurement**
    - Send an ultrasonic pulse by setting the trigger pin HIGH for 10 microseconds and then LOW.
    - Start a timer and wait for the echo signal (received at the echo pin) to go HIGH.
    - Measure the duration of the HIGH pulse using the timer (duration corresponds to the time taken by the ultrasonic wave to travel to the object and back).
5. **Calculate Distance**
    - Use the formula to calculate the distance:
      Distance (cm)=Time (microseconds)×Speed of Sound (340 m/s)2×104\text{Distance (cm)} = \frac{\text{Time (microseconds)} \times \text{Speed of Sound (340 m/s)}}{2 \times 10^4}Distance (cm)=2×104Time (microseconds)×Speed of Sound (340 m/s)
        - Divide the time by 2 because the ultrasonic wave travels to the object and back.
6. **Display Output**
    - Convert the measured distance into a suitable format and update the output display.
    - If no object is detected within the sensor's range, display "Out of Range" or a similar message.
7. **Return to Idle State**
    - Wait for a brief interval to avoid rapid re-triggering, then return to monitoring the touch sensor for new activation.
8. **Repeat the Process**
    - Continuously loop through the above steps, ensuring the system is always ready for touch activation.

Chapter 4

RESULTS





Chapter 5

Applications

1. **Industrial Automation**
   - Enables real-time object detection and distance measurement for assembly lines and warehouses.
2. **Robotics**
   - Used for obstacle detection, navigation, and proximity sensing in autonomous and interactive robots.
3. **Parking Assistance Systems**
   - Assists in measuring the distance to obstacles for safe and accurate vehicle parking.
4. **Smart Home Automation**
   - Facilitates touch-activated systems for tasks like controlling motorized doors, curtains, or occupancy detection.
5. **Assistive Technology**
   - Provides a tool for visually impaired individuals to measure distances or avoid obstacles.
6. **Healthcare Devices**
   - Integrated into medical equipment for distance measurement in diagnostic or therapeutic applications.
7. **Agriculture**
   - Measures distances for precision farming, such as plant spacing or water level monitoring.

Conclusion

Touch sensors, ultrasonic sensors, and microcontroller hardware are all seamlessly integrated in the Touch Activated Distance Measurement system that uses register-level programming. The project guarantees effective control, accurate timing, and optimal resource utilisation by utilising register-level programming. By only turning on the system when necessary, the touch-based activation function improves usability and energy economy and makes it suitable for a variety of applications, including assistive technology, automation, and robotics.

The importance of low-level hardware-software interactions in creating effective embedded systems is demonstrated by this effort. It provides a useful learning environment for comprehending real-time system design, sensor interface, and microcontroller programming. In addition, the system's flexibility and scalability create opportunities for deployment across a range of industries, functioning as a basis for the creation of cutting-edge touch-interactive systems.