EC2094D

MICRO PROCESSOR MICROCONTROLLER LAB

MPMC PROJECT REPORT

NAME	ROLLNO	EMAIL
Kadagala Shiva Kumar	B200981EC	kadagala_b200981ec@nitc.ac.in

Project Report

Ultrasonic Rangefinder using 8051 Microcontroller

Objective :

The objective of this experiment is to build a Ultrasonic rangefinder using 8051 Microcontroller and Ultrasonic Sensor which measures the distance up to 4 meters.

1. Introduction: The Ultrasonic Rangefinder project is designed to measure the distance of an object using the principle of SONAR. It employs the HC-SR04 Ultrasonic Module, an efficient and accurate range detection sensor, interfaced with the 8051 microcontroller. The 8051 microcontroller calculates the distance based on the time taken for the ultrasonic signal to travel to the object and back. The calculated distance is then displayed on an LCD screen.

2. Working Principle: The Ultrasonic Module generates eight pulses of 40 kHz when triggered. After sending the pulses, it waits for the echo signal to return. The time duration for which the echo signal remains high is directly proportional to the distance of the object. Using this time duration and the speed of sound in air, the microcontroller calculates the distance using the formula:

1. **Object Distance (in cm) = (Sound Velocity * Time) / 2**

   where Sound Velocity is 34300 cm/s at normal temperature.

2. **Here, oscillator frequency of AT89S52 (8051) is 11.0592 MHz, then timer frequency of 8051 will be 921.6 kHz. 3.**
3. **So, Time required to execute 1 instruction is 1.085 us. 4.**
4. **So, timer gets incremented after 1.085 us time elapse.**

Hence,

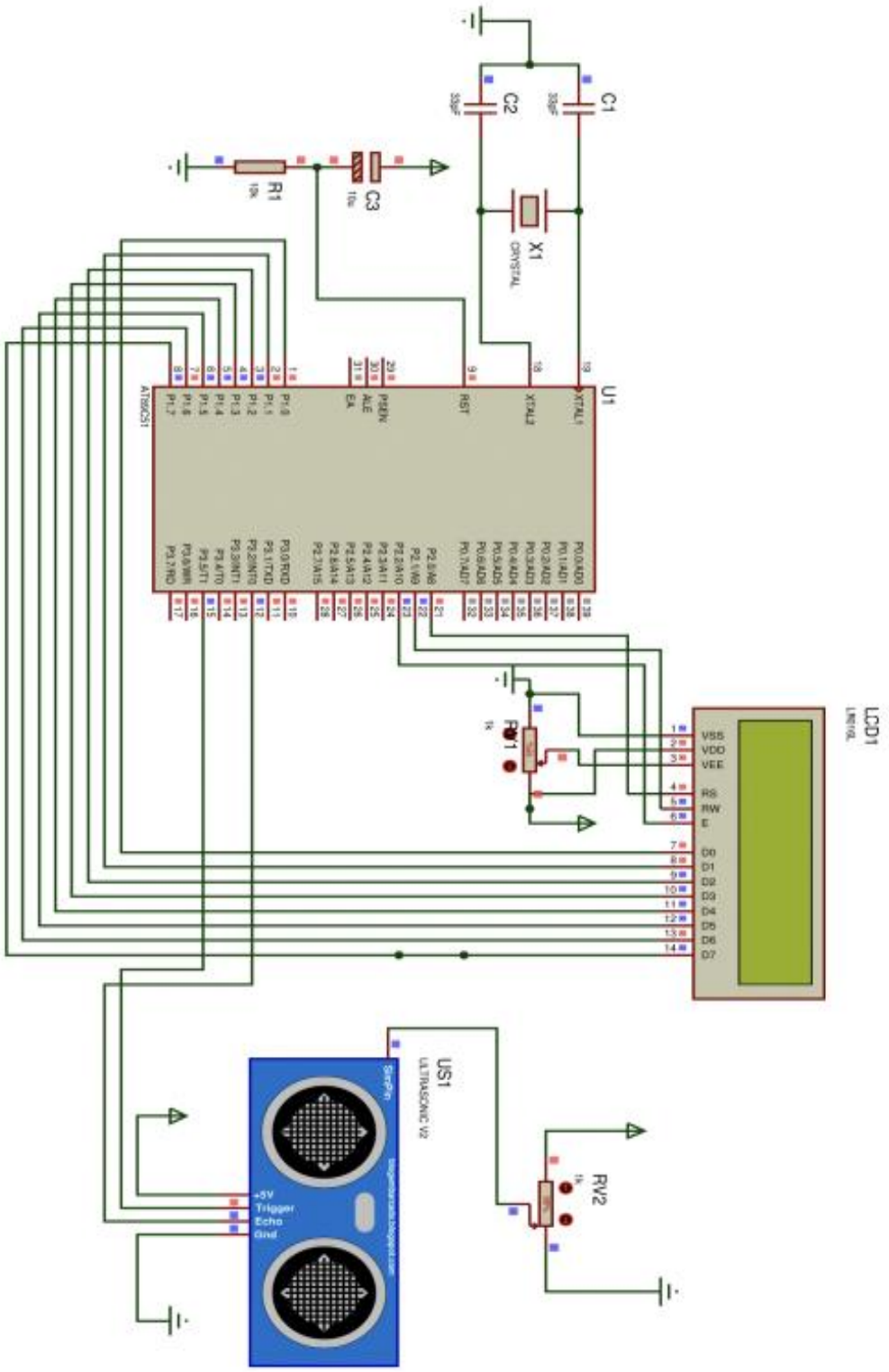
$$\text{Distance} = (34300 * \text{TimerCount} * 1.085 * 10^{-6})/2$$

$$= \text{TimerCount}/54$$

3. Components Used:

8051 Microcontroller (AT89S52)
HC-SR04 Ultrasonic Module
LCD Display (compatible with HD44780 controller)
Resistors and Capacitors for timing and interfacing
Power supply (5V DC)

Interfacing Diagram:



4. Project Description: The project starts by initializing the LCD display with proper configurations for 2 lines and 5x7 matrix characters. It then clears the display and sets up the cursor for further display operations.The ultrasonic module is triggered using Timer 0, which is configured in 16-bit mode with gate enable. The microcontroller measures the time taken by the ultrasonic signal to travel to the object and back by reading Timer 0's count.
Once the time is calculated, it is converted to distance using the formula mentioned above. The distance is then converted to a 4-digit ASCII number for display on the LCD.

5. Procedure:

Burn the project code to the 8051 microcontroller.
Connect the HC-SR04 Ultrasonic Module to the specified pins on the microcontroller.
Connect the LCD display to the microcontroller's data and control pins.
Supply power to the circuit (5V DC).
Place an obstacle in front of the ultrasonic module to measure the distance.
Observe the measured distance displayed on the LCD screen.
The distance will be updated continuously due to the infinite loop in the program.
To stop the operation, switch off the power supply.

Code:

```
; Header file inclusions for 8051
#include <reg51.h>
#include <intrins.h>

; Define constants for LCD control pins
LCDrs EQU P2.0 ; Register select Pin
LCDrw EQU P2.1 ; Read/Write Pin
LCDen EQU P2.2 ; Enable Pin

; Define constants for sensor pins
trig EQU P3.5 ; Timer 1
echo EQU P3.2 ; INT0

; Data memory variables
range DB 0 ; Variable to store the measured range

; Code memory variables (to store strings)
```

```
str1 DB "OBSTACLE  AT ", 0
str2 DB "0000 CM", 0

; Code memory constants
TIMER_DELAY EQU 1275 ; Delay value for the delay function

; Code memory initialization
ORG 0 ; Start the code at address 0

MAIN:
    ; Initialize LCD
    MOV A, #0x30
    CALL COMMAND ; 1 line and 5x7 matrix
    CALL DELAY

    MOV A, #0x38
    CALL COMMAND ; 2 lines and 5x7 matrix
    CALL DELAY

    MOV A, #0x0C
    CALL COMMAND ; Display on, cursor off
    CALL DELAY

    MOV A, #0x01
    CALL COMMAND ; Clear display Screen
    CALL DELAY

    MOV A, #0x06
    CALL COMMAND ; Shift cursor to right
    CALL DELAY
```

```
    MOV DPTR, #str1
    CALL DISPLAY_LCD ; Display "OBSTACLE  AT "

    ; Timer 0 initialization
    MOV TMOD, #0x09 ; Timer0 in 16-bit mode with gate enable
    SETB TR0 ; Timer run enabled
    MOV TH0, #0x00
    MOV TL0, #0x00

    SETB echo ; Set echo pin (P3.2) as input

LOOP:
    CALL GET_RANGE
    CALL DELAY ; Delay for 2 ms
    SJMP LOOP ; Infinite loop

; Subroutine to generate delay
DELAY:
    MOV R1, #TIMER_DELAY
DELAY_LOOP:
    DJNZ R1, DELAY_LOOP
    RET

; Subroutine to send commands to the LCD
COMMAND:
    CLR LCDrs
    CLR LCDrw
    SETB LCDen ; Strobe the enable pin
    MOV P1, A ; Put the value on the pins
    CLR LCDrs
    CLR LCDrw
```

```
    CLR LCDen
    RET

; Subroutine to display a string on the LCD
DISPLAY_LCD:
    MOV R1, A ; Load the address of the string
DISPLAY_LOOP:
    MOV A, @R1 ; Load the character from the code memory
    CJNE A, #0, DISPLAY_CHAR ; If not the null terminator, display the character
    RET ; End of string, return

DISPLAY_CHAR:
    SETB LCDrs ; Set RS to 1 for data mode
    CLR LCDrw
    SETB LCDen ; Strobe the enable pin
    MOV P1, A ; Put the character on the pins
    CLR LCDrs
    CLR LCDrw
    CLR LCDen
    INC R1 ; Move to the next character in the string
    ACALL DELAY ; 10 ms delay
    SJMP DISPLAY_LOOP ; Continue displaying the string

; Subroutine to measure the range using ultrasonic sensor
GET_RANGE:
    ; Send the pulse
    CLR TR0 ; Stop Timer0
    MOV TH0, #0x00
    MOV TL0, #0x00
    SETB trig ; Pull trigger pin (P3.5) HIGH
    NOP ; Delay for trigger pulse
```

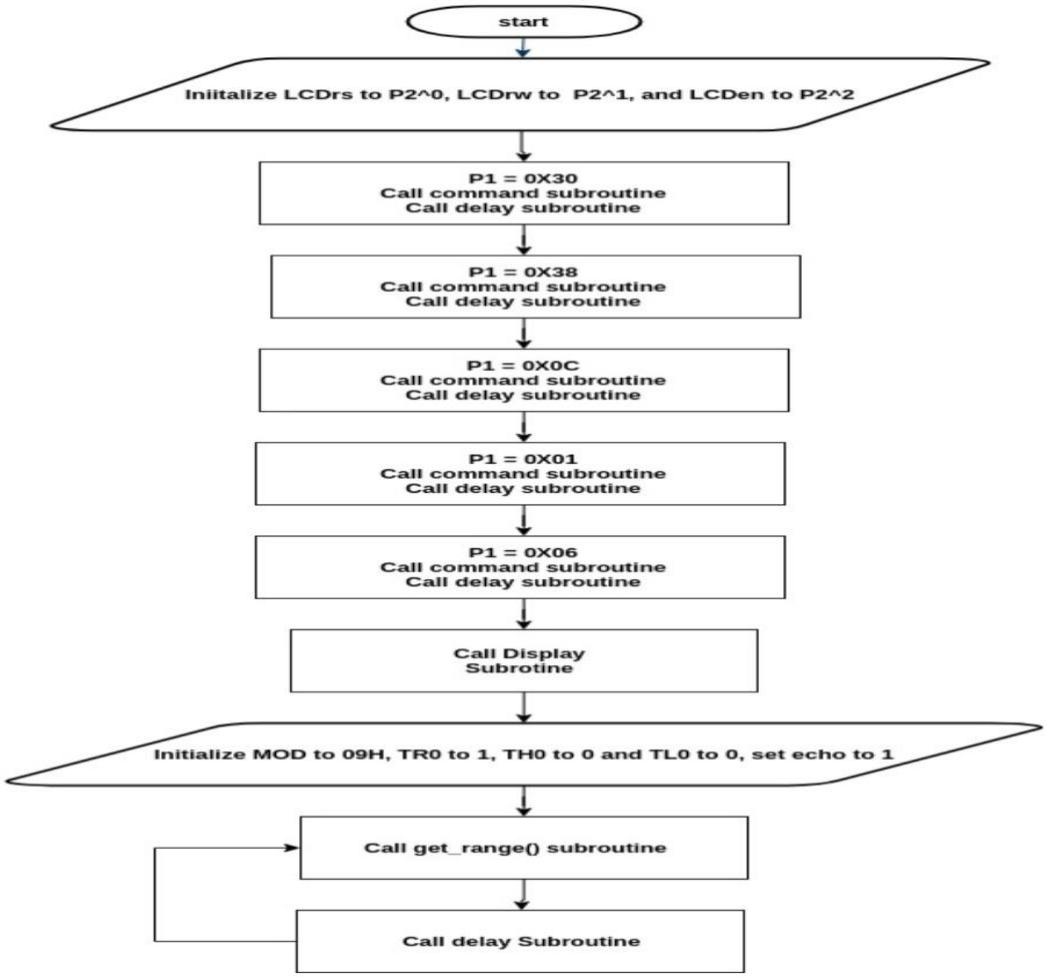
```
    NOP
    NOP
    NOP
    NOP
    CLR trig ; Pull trigger pin LOW

WAIT_FOR_ECHO:
    JB echo, WAIT_FOR_ECHO ; Wait until echo pulse is detected
    JNB echo, WAIT_FOR_ECHO ; Wait until echo changes its state

    ; Read the timer value
    MOV A, TH0
    MOV B, TL0
    MOV R7, A ; Store high byte in R7
    MOV R6, B ; Store low byte in R6
    MOV A, R7
    MOV B, R6
    ACALL CALCULATE_RANGE ; Calculate the range
    RET

; Subroutine to calculate and display the range
CALCULATE_RANGE:
    MOV A, R7 ; High byte of timer value
    MOV B, R6 ; Low byte of timer value

    ; Convert the timer value to microseconds
    MOV R0, #54 ; 1 clock cycle is 1/12 MHz = 83.3 ns, 1 us = 12 clock cycles
    MUL AB ; Multiply high and low bytes by 54
    MOV R2, A ; Store the result in R2
    MOV R3, B
```

```
    MOV A, #0 ; Clear accumulator

    ; Check if the range is less than 34300 microseconds (34.3 ms)
    MOV R1, #0xD4 ; Load the threshold value (34300 / 1000 = 34.3 ms)
    CJNE R2, R1, RANGE_OK ; If range is less than 34.3 ms, skip the division
    CJNE R3, #0x2C, RANGE_OK

    ; Range is greater than or equal to 34.3 ms, set range to 0
    SJMP DISPLAY_RANGE

RANGE_OK:
    ; Divide the timer value by 54 to get the range in centimeters
    MOV A, R2 ; High byte of multiplied value
    MOV B, R3 ; Low byte of multiplied value
    MOV R2, #0 ; Clear R2 and R3 for division
    MOV R3, #0
    MOV R4, #54 ; Divisor for division
    DIV AB ; Divide R1R0 by R3R2
    MOV A, R0 ; Quotient (range in centimeters) in accumulator

    ; Convert the range to a 4-digit number in ASCII
    MOV R5, #4 ; Number of digits (4 digits)
CONVERT_TO_ASCII:
    MOV A, #0x30 ; ASCII '0'
    ADD A, R0 ; Add the remainder to '0'
    MOV R0, A ; Update R0 with the ASCII digit
    MOV A, R5 ; Load the digit position
    ADD A, #0x0C ; Add the ASCII offset for the LCD display
    MOV @R0, A ; Store the digit in the output string
    DEC R5 ; Move to the next digit
    DJNZ CONVERT_TO_ASCII ; Repeat for all digits
```

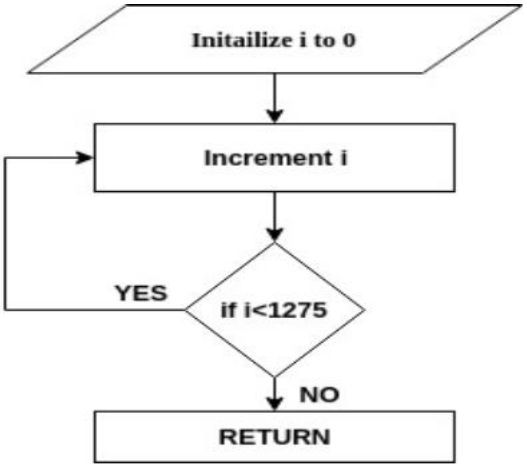
```
DISPLAY_RANGE:
    ; Display the range on the LCD
    MOV DPTR, #str2 ; Point DPTR to the output string
    CALL DISPLAY_LCD ; Display the string

    RET ; Return from the subroutine
```

6.FlowChart:

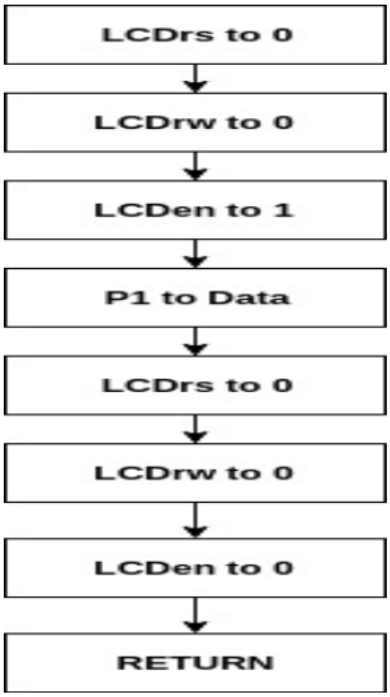
1.Main Function



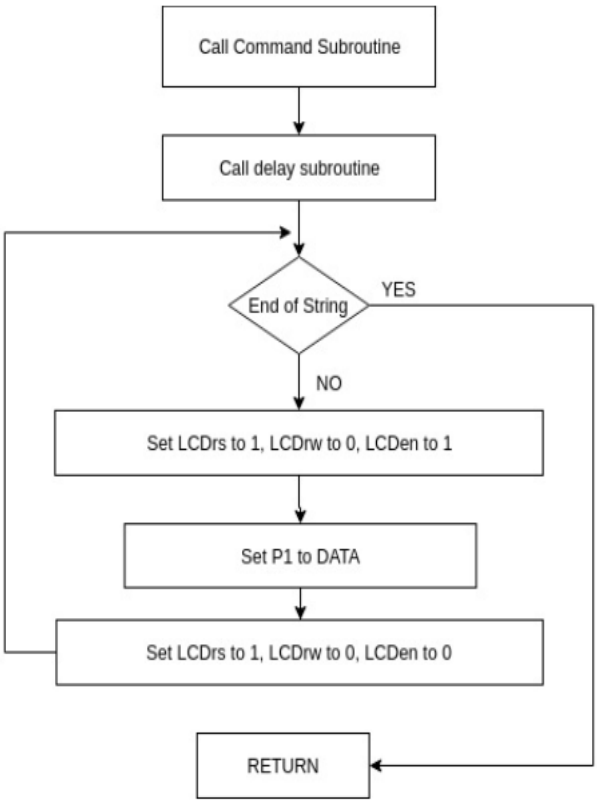
2.Delay



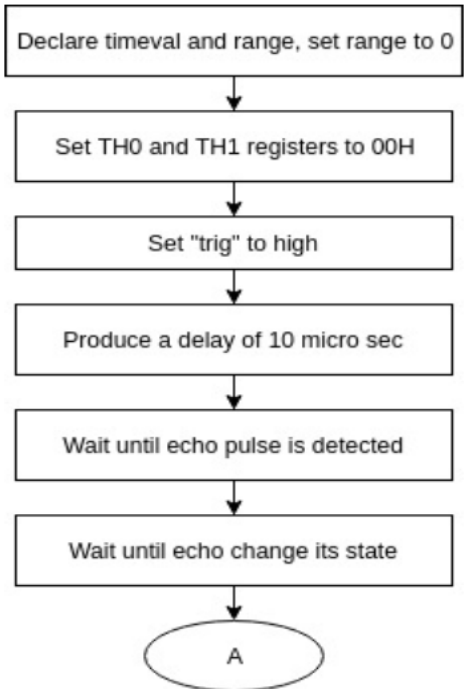
3.Command

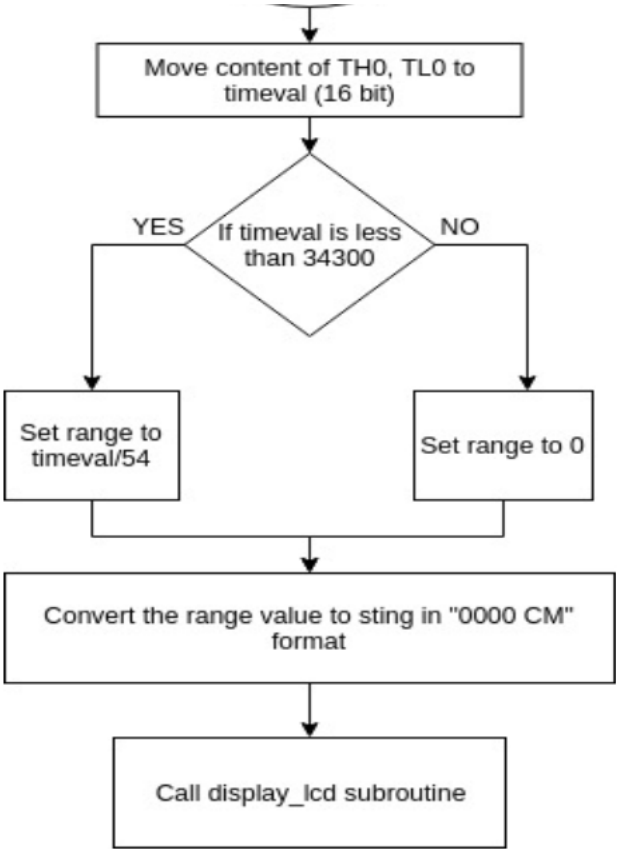


4.Dispaly LCD:


Call Command Subroutine
Call delay subroutine
End of String
YES
NO
Set LCDrs to 1, LCDrw to 0, LCDen to 1
Set P1 to DATA
Set LCDrs to 1, LCDrw to 0, LCDen to 0
RETURN

5.Get Range:


Declare timeval and range, set range to 0
Set TH0 and TH1 registers to 00H
Set "trig" to high
Produce a delay of 10 micro sec
Wait until echo pulse is detected
Wait until echo change its state
A



7. Results: The Ultrasonic Rangefinder project has been successfully implemented and tested. It accurately measures the distance of an object using the ultrasonic module and displays the results on the LCD screen. The project can measure distances up to 4 meters with high accuracy and stable readings.

8. Conclusion: The Ultrasonic Rangefinder using the 8051 microcontroller is an effective distance measurement system. The project demonstrates the principles of SONAR and effectively calculates the distance based on time-of-flight measurements. The implementation is robust and provides accurate distance readings. The project could be further improved by adding features like calibration, user interface, and enhanced display functionalities.

9.Ultrasonic Rangefinder Applications:

1. Used to measure the obstacle distance.
2. This system used in automotive parking sensors and obstacle warning systems.
3. Used in terrain monitoring robots.

10. Recommendations:

Ensure proper connections and configurations while setting up the circuit.
Implement appropriate safety precautions when handling electrical components.
Consider calibrating the system to improve accuracy.
Implement additional features like distance unit selection and range threshold alerts for specific applications.

11. Future Enhancements:

Implement a user interface for better user interaction and control.
Integrate a data logging feature to record distance measurements.
Enhance the LCD display to show additional information and sensor status.
Incorporate wireless communication to transmit distance data to remote devices.
The Ultrasonic Rangefinder project serves as an excellent example of practical distance measurement applications using microcontrollers. With further development, it can find applications in robotics, automated systems, and smart devices.