SmartPark Distance Guardian

Project Synopsis Submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

Of

Bachelor of Technology

in

Information Technology

By

Pavit Agrawal	**Ishan Saraf**	**Bhavya Khandelwal**
230953406	**230953404**	**230953408**

Under the guidance of

Dr. Santosha Kamath (Lab Faculty 1)
Associate Professor – Senior Scale
School of Computer Engineering
Manipal Institute of Technology
MAHE, Manipal, Karnataka, India

Dr. Anirudhan (Lab faculty 2)
Assistant Professor –Senior Scale
School of Computer Engineering
Manipal Institute of Technology
MAHE, Manipal, Karnataka, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

October 2025

TABLE OF CONTENTS

I. ABSTRACT

Our project, the "SmartPark Distance Guardian", aims to introduce a solution to improve parking safety through cutting-edge technology. It uses an ARM microcontroller kit and ultrasonic sensors to create a parking distance monitoring system. This project addresses the need for a dependable, user-friendly solution to ensure secure parking practices in crowded environments.

Using ultrasonic sensors to measure precise distances between parked cars, implementing an intelligent alert system with LED indicators for distances less than three feet, and creating an easy-to-use user interface to provide users with real-time feedback are some of the main goals of the " SmartPark Distance Guardian". The system demonstrates its reliability and usability across various parking scenarios through meticulous testing and calibration.

Sustainable Development Goal (SDG):

Aligned with Sustainable Development Goal 11: "Sustainable Cities and Communities," the "SmartPark Distance Guardian" creates safer and more sustainable urban environments. By promoting secure parking practices and minimizing the risk of vehicle damage in congested areas, it supports efficient utilization of parking spaces, reduces congestion, and enhances overall safety in city settings. This endeavor towards safer parking solutions aligns with creating sustainable and resilient cities for all

II. INTRODUCTION

2.1 Scope:

The project aims to design a parking distance monitoring system utilizing an ultrasonic sensor, providing real-time parking distance information displayed on an LCD screen. A built-in LED indicator will signal when distances fall below 3 feet, endorsing safe parking and ensuring public safety in congested parking environments.

2.2 Project Description:

The "SmartPark Distance Guardian" is an innovative parking distance monitoring system meticulously engineered to optimize the parking experience in busy parking scenarios. The device uses advanced ultrasonic sensor technology to accurately measure the distance between parked vehicles in real time. The obtained parking distance data is immediately exhibited on an LCD screen, offering users immediate insights. An intelligent alert system featuring an LED indicator activates when the measured distance between vehicles descends below the recommended 3 feet, promoting secure parking practices. This system supports continuous monitoring, making it suitable for various applications, including commercial parking lots, public garages, and residential parking spaces. By streamlining the utilization of parking spaces and guaranteeing that vehicles are parked at a secure distance from one another, the "SmartPark Distance Guardian" improves both the convenience and safety of parking, ultimately decreasing the risk of minor vehicle damage and enhancing the overall parking experience.

2.3 Problem Statement:

Develop a "SmartPark Distance Guardian" with ultrasonic sensors for real-time parking distance on an LCD screen. Include an LED alert for distances below 3 feet, ensuring safe parking and preventing vehicle damage in crowded areas.

2.4 Objective:

The primary goal of the "SmartPark Distance Guardian" project is to establish a dependable and user-friendly parking distance monitoring system. This system employs an ultrasonic sensor to gauge the distance between parked vehicles in real-time precisely. The collected distance data is promptly presented on the LCD screen of the ARM microcontroller kit, delivering instantaneous feedback to users. Furthermore, the system is equipped with an intelligent, alert system: an LED indicator activates whenever the measured distance dips below the recommended safety threshold of 3 feet. By blnding cutting-edge technology with user safety, the SmartPark Distance Guardian aims to encourage secure parking and enhance the convenience and safety of parking in crowded situations.

III. LITERATURE SURVEY

The SmartPark Distance Guardian project is anchored in the pursuit of innovative parking safety solutions, drawing inspiration from diverse explorations in the realm of proximity monitoring.

Benefits of using SmartPark Distance Sensors :

By providing accurate information about available parking space, the proximity decrease unnecessary fuel consumption and accident caused by drivers while parking, thereby contributing to reduced resource wastage and safety. **[3]**

Operating Principles of Ultrasonic Sensors:

Ultrasonic sensors function by emitting high-frequency waves beyond human auditory range and analyze their reflections. It is comprised of a transmitter and receiver, these sensors use a piezoelectric crystal to emit and interpret ultrasonic signals. **[1]**

Comparative Analysis of Ultrasonic Sensor Methods:

In ultrasonic-range sensors, the pulse-echo method and continuous-wave method are significant. While the latter demands more expensive equipment, the former, particularly utilizing low-cost sensors, offers a cost-effective alternative. The paper highlights the inherent limitations in precision when employing threshold-detection techniques with such sensors. **[2]**

Ultrasonic Sensors and LED Guidance in Parking Systems:

Ultrasonic sensors function using echolocation, emitting sound waves that bounce off and return to the sensor. A LED display board assists the driver, with reflections altered in timing when a vehicle is detected. This setup shares similarities with a typical Informative Parking System (IPS). **[3]**

IV. SYSTEM REQUIREMENTS

4.1 Hardware Requirements:

The following components have been used:

1. **ALS-SDA-ARMCTXM3-01**: ARM Cortex-M3 Development Board - Utilized as the central microcontroller for processing and controlling the SmartPark Distance Guardian system.

2. **Power supply (+5V)**: Provides the necessary voltage to power the ARM Cortex-M3 board and associated components, ensuring proper functionality.

3. **Cross-cable:** Facilitates programming and serial communication between devices, aiding in software uploads and data transfer.

4. **One working USB port on the host computer system and PC**: Essential for software download and transfer from the host computer to the ARM Cortex-M3 board.

5. **10 core FRC cables of 8-inch length**: Used for internal connections and wiring within the SmartPark Distance Guardian system, ensuring efficient signal transmission between components.

6. **HC-SR04 Ultrasonic Distance Sensor**: The HC-SR04 distance sensor employs ultrasonic technology for precise non-contact distance measurements in diverse sonar applications. Its optimal operating range is 2 cm to 400 cm, with an accuracy level of up to 0.5cm.

7. **USB to B type cable**: Enables connectivity between devices and facilitates data transfer, particularly useful for linking the ARM Cortex-M3 board with other peripherals or the host computer system.

4.2 Software Requirements:

- **Language: Embedded C** : Used to program the microcontroller that controls the SmartPark Distance Guardian.
- **IDE : Keil MicroVision** : Used to develop and debug the embedded C code for the SmartPark Distance Guardian.
- **Application: Flash Magic** : Used to program the microcontroller with the SmartPark Distance Guardian firmware

V. FUNCTIONALITY [1]

Working Principle:

The HC-SR04 Ultrasonic Distance/Ranging Sensor utilizes ultrasound to gauge the distance to an object. Ultrasound, characterized by a frequency beyond the audible range (> 20 kHz), forms the basis of its operation. With a detection range spanning 2 centimeters to 4 meters, the HCSR04 module employs a 40 kHz ultrasound signal to measure the distance between itself and any object within its field.

Pinout: In terms of pinout, the module features four pins: VCC (+5V), TRIG, ECHO, and GND.

Like SONAR, the ultrasonic sensor incorporates two transducers—one for transmitting ultrasound and the other for receiving the echo. Distance calculation relies on the speed of sound in the air, set at 343 m/s.

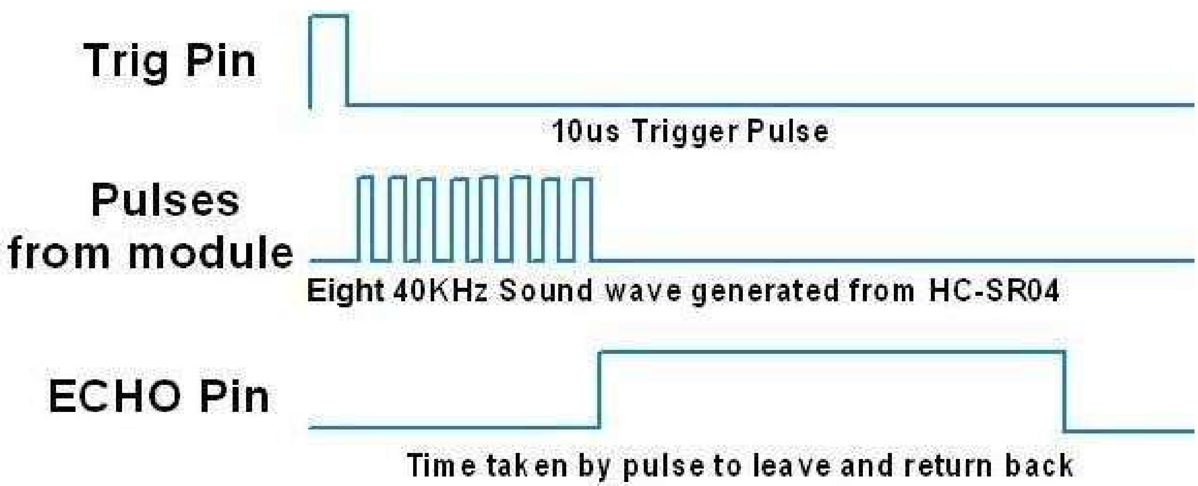

Fig 1. Ultrasonic HC-SR04 module timing diagram

Interfacing HC-SR04:

1. The trigger pin is given a short pulse of 10us.
2. Upon receiving a trigger pulse, the HC-SR04 Module emits a burst of eight ultrasonic pulses at 40 kHz.
3. Then, it outputs a HIGH for the time the sound waves take to reach back.
4. The duration of the high pulse is measured and subsequently utilized to determine the distance.

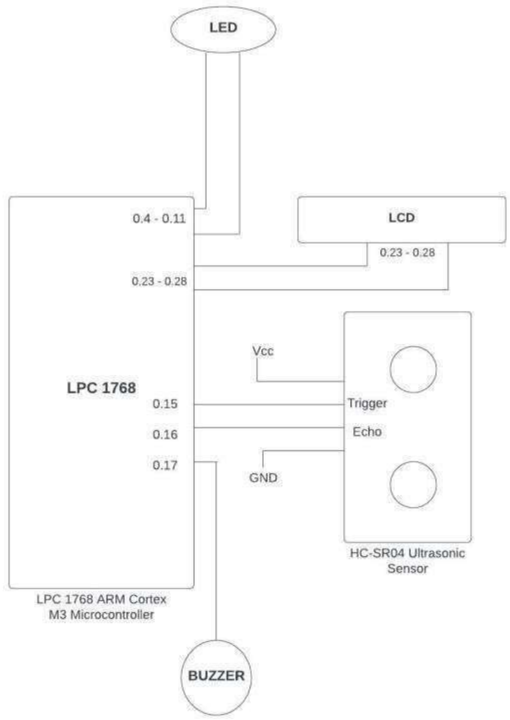

Fig 2. Circuit Diagram: LPC1768 interfaced with HC-SR04 ultrasonic sensor, LEDs, LCD, and buzzer, displaying the respective pin configurations.

Distance Measurement Calculations:

The speed of sound in the air is typically 343 meters per second. To determine the distance traveled by sound waves, we employ the formula: Distance traveled (DT) in centimeters equals 0.00343 (mm/µs) multiplied by the time in microseconds (µs).

After obtaining the result in centimeters, we account for the round trip of the sound waves by dividing the distance by 2. Therefore, the final formula is expressed as D = 0.0343 times T/2 in centimeters.

In summary, this formula facilitates the calculation of the distance covered by a sound wave, considering the speed of sound and the time taken for the sound to travel to a surface and back. The division by 2 accommodates the round-trip nature of the distance. This method is advantageous when measuring time in microseconds and expressing the distance in centimeters.

VI. CODE

```
#include <stdio.h>
#include <LPC17xx.h>
#include <string.h>

#define LED_Pinsel 0xFF        // P0.4–P0.11 (LEDs)
#define TRIGGER_PIN (1 << 15)  // P0.15 (Trigger)
#define ECHO_PIN (1 << 16)     // P0.16 (Echo)

// New additional GPIO pins
#define EXT_LED_P1_MASK ((1 << 23) | (1 << 24) | (1 << 25) | (1 << 26))
#define EXT_LED_P2_MASK ((1 << 10) | (1 << 11) | (1 << 12) | (1 << 13))

char ans[20] = "";
int temp, temp1, temp2 = 0;
int flag = 0, flag_command = 0;
int i, j, k, l, r, echoTime = 5000;
float distance = 0;
int b;

void lcd_wr(void);
void port_wr(void);
void delay(int r1);
void timer_start(void);
float timer_stop(void);
void timer_init(void);
void delay_in_US(unsigned int microseconds);

// ---------------- Delay in Microseconds ----------------
void delay_in_US(unsigned int microseconds) {
   LPC_TIM0->TCR = 0x02;                // Reset Timer
   LPC_TIM0->PR = 0;                    // Prescaler = 0
   LPC_TIM0->MR0 = microseconds - 1;    // Match register
   LPC_TIM0->MCR = 0x01;                // Interrupt on match
   LPC_TIM0->TCR = 0x01;                // Start timer
   while ((LPC_TIM0->IR & 0x01) == 0);  // Wait for interrupt flag
   LPC_TIM0->TCR = 0x00;                // Stop timer
   LPC_TIM0->IR = 0x01;                 // Clear flag
}

// ---------------- Timer Initialization ----------------
void timer_init(void) {
   LPC_TIM0->CTCR = 0x0;
   LPC_TIM0->PR = 11999999;   // 12 MHz base
   LPC_TIM0->TCR = 0x02;      // Reset timer
}
```

```
// ---------------- Timer Start ----------------
void timer_start(void) {
   LPC_TIM0->TCR = 0x02;  // Reset
   LPC_TIM0->TCR = 0x01;  // Enable
}

// ---------------- Timer Stop ----------------
float timer_stop(void) {
   LPC_TIM0->TCR = 0x0;
   return LPC_TIM0->TC;
}

// ---------------- Simple Software Delay ----------------
void delay(int r1) {
   for (r = 0; r < r1; r++);
}

// ---------------- LCD Write Port ----------------
void port_wr(void) {
   int j;
   LPC_GPIO0->FIOPIN = temp2 << 23;
   if (flag_command == 0)
      LPC_GPIO0->FIOCLR = 1 << 27;
   else
      LPC_GPIO0->FIOSET = 1 << 27;

   LPC_GPIO0->FIOSET = 1 << 28;
   for (j = 0; j < 50; j++);
   LPC_GPIO0->FIOCLR = 1 << 28;
   for (j = 0; j < 10000; j++);
}

// ---------------- LCD Write ----------------
void lcd_wr(void) {
   temp2 = (temp1 >> 4) & 0xF;
   port_wr();
   temp2 = temp1 & 0xF;
   port_wr();
}

// ---------------- MAIN FUNCTION ----------------
int main(void) {
   int command_init[] = {3, 3, 3, 2, 2, 0x01, 0x06, 0x0C, 0x80};

   SystemInit();
   SystemCoreClockUpdate();
   timer_init();
```

```
    // GPIO Configurations
    LPC_PINCON->PINSEL0 &= 0xFFFFF00F; // P0.4–P0.11 as GPIO (LEDs)
    LPC_PINCON->PINSEL0 &= 0x3FFFFFFF; // P0.15 as GPIO (Trigger)
    LPC_PINCON->PINSEL1 &= 0xFFFFFFF0; // P0.16 as GPIO (Echo)

    // Clear all function bits for the new extended LED pins
    LPC_PINCON->PINSEL3 &= ~((3 << 14) | (3 << 16) | (3 << 18) | (3 << 20)); // P1.23–P1.26
GPIO
    LPC_PINCON->PINSEL4 &= ~((3 << 20) | (3 << 22) | (3 << 24) | (3 << 26)); // P2.10–P2.13
GPIO

    LPC_GPIO0->FIODIR |= TRIGGER_PIN;       // Trigger as Output
    LPC_GPIO0->FIODIR &= ~ECHO_PIN;         // Echo as Input
    LPC_GPIO0->FIODIR |= LED_Pinsel << 4;   // P0.4–P0.11 LEDs Output
    LPC_GPIO0->FIODIR |= 0xF << 23 | 1 << 27 | 1 << 28; // LCD Pins Output
    LPC_GPIO0->FIODIR |= 1 << 17;           // Indicator LED
    LPC_GPIO1->FIODIR |= EXT_LED_P1_MASK;   // P1.23–P1.26 as Output
    LPC_GPIO2->FIODIR |= EXT_LED_P2_MASK;   // P2.10–P2.13 as Output

    // LCD Initialization
    flag_command = 0;
    for (i = 0; i < 9; i++) {
        temp1 = command_init[i];
        lcd_wr();
        for (j = 0; j < 30000; j++);
    }

    LPC_GPIO0->FIOCLR |= TRIGGER_PIN;

    while (1) {
        // Trigger Pulse (10 µs)
        LPC_GPIO0->FIOSET = TRIGGER_PIN;
        delay_in_US(10);
        LPC_GPIO0->FIOCLR = TRIGGER_PIN;

        // Wait for Echo High
        while (!(LPC_GPIO0->FIOPIN & ECHO_PIN));
        timer_start();
        // Wait for Echo Low
        while (LPC_GPIO0->FIOPIN & ECHO_PIN);
        echoTime = timer_stop();

        // Distance Calculation
        distance = (0.00343 * echoTime) / 2; // in cm
        sprintf(ans, " Distance: %.2f cm", distance);

        // Clear LCD and Display Distance
        flag_command = 0;
```

```
        temp1 = 0x01;
        lcd_wr();

        flag_command = 1;
        i = 0;
        while (ans[i] != '\0') {
            temp1 = ans[i];
            lcd_wr();
            for (j = 0; j < 300000; j++);
            i++;
        }

        // ---------------- LED BLINK CONTROL ----------------
        if (distance < 20) {
            // Faster, brighter blink for close object
            for (b = 0; b < 10; b++) {
                // All LED groups ON
                LPC_GPIO0->FIOSET = LED_Pinsel << 4; // P0 LEDs ON
                LPC_GPIO1->FIOSET = EXT_LED_P1_MASK; // P1 LEDs ON
                LPC_GPIO2->FIOSET = EXT_LED_P2_MASK; // P2 LEDs ON
                delay(30000);                        // ON ? brighter

                // All LED groups OFF
                LPC_GPIO0->FIOCLR = LED_Pinsel << 4;
                LPC_GPIO1->FIOCLR = EXT_LED_P1_MASK;
                LPC_GPIO2->FIOCLR = EXT_LED_P2_MASK;
                delay(5000);                         // OFF ? faster blink
            }
            LPC_GPIO0->FIOSET = 1 << 17; // Indicator LED ON
        } else {
            // LEDs OFF when object far
            LPC_GPIO0->FIOCLR = LED_Pinsel << 4;
            LPC_GPIO1->FIOCLR = EXT_LED_P1_MASK;
            LPC_GPIO2->FIOCLR = EXT_LED_P2_MASK;
            LPC_GPIO0->FIOCLR = 1 << 17;
        }

        // General delay between sensor readings
        delay(60000);
    }
}
```

VII. RESULTS

When the ultrasonic sensor detects a distance > 20 cm, the system measures the distance between sensor and the object and displays it on the LCD screen, as depicted in Figure 3. Conversely, when the detected distance is < 20 cm, the system promptly measures the distance and displays it on the LCD screen. Additionally, as an alert mechanism, the system initiates a continuous beeping sound from the buzzer as long as the object remains within this proximity.

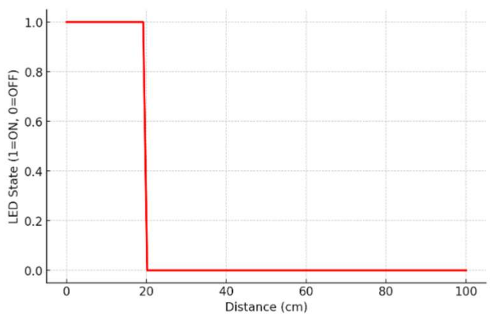

Fig 3. Distance vs LED Activation

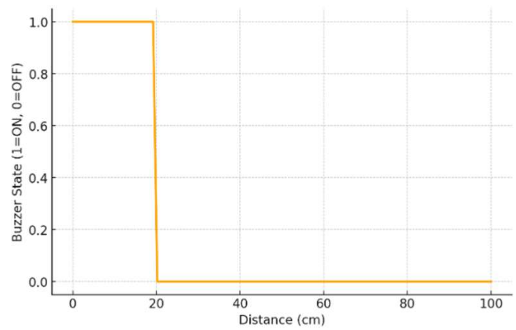

Fig 4. Distance vs Buzzer Activation

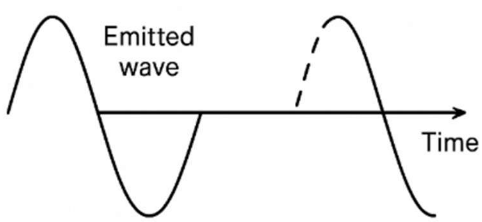

Fig 5. Ultrasonic Emission & Echo

Fig 6. Amplitude vs Distance Decay

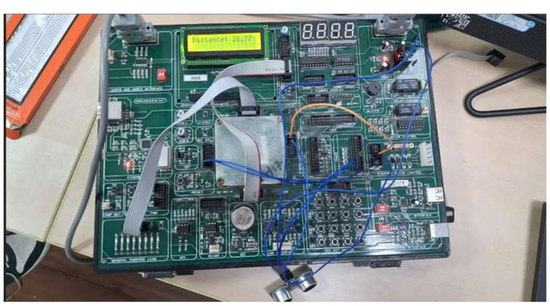

Fig 7. Observation when distance>20cm

Fig 8. Observation when distance<20cm

VIII. CONCLUSION

In conclusion, the successful development and implementation of the "SmartPark Distance Guardian" project mark a significant achievement in addressing the pressing need for secure and efficient parking in congested environments. By leveraging advanced technology, including an ARM microcontroller kit and ultrasonic sensors, the system provides real-time monitoring of parking distances, contributing to enhanced safety for both vehicles and pedestrians.

The integration of an intelligent LED alert system, activated when distances fall below the recommended safety threshold, serves as a proactive measure to prevent potential collisions and vehicle damage. The user-friendly interface displayed on the LCD screen of the ARM microcontroller kit ensures that drivers receive instant feedback, promoting informed and safe parking practices.

The project's alignment with Sustainable Development Goal 11 underscores its commitment to creating safer and more sustainable urban environments. By optimizing parking space utilization and mitigating collision risks, the "SmarkPark Distance Guardian" contributes to the development of resilient and secure cities.

In summary, the "SmartPark Distance Guardian" not only represents a significant step forward in parking distance monitoring technology but also stands as a tangible solution to the challenges associated with crowded parking scenarios. This innovation enhances the overall parking experience, reduces the risk of minor vehicle damage, and fosters a safer and more sustainable urban landscape for all.

IX. REFERENCES

[1] Robocraze. (n.d.). What is Ultrasonic Sensor: Working Principle & Applications. No Author info. Retrieved from https://robocraze.com/blogs/post/what-is-ultrasonic-sensor -header-15
Referred on: 15th October, 2025

[2] Aliew, F. (2022). An Approach for Precise Distance Measuring Using Ultrasonic Sensors. *Eng. Proc.*, *24*(1), 8. Retrieved from https://doi.org/10.3390/IECMA2022-12901
Referred on: 15th October, 2025

[3] Sahoo, Biren & Shahjad, & Tanwar, Prakash. (2021). Real-Time Smart Parking: Challenges and Solution using Machine learning and IoT. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 451-458. 10.32628/CSEIT217295.Retrived from https://doi.org/10.32628/CSEIT217295
Referred on: 15th October, 2025

ESD_Report_CCEB_B2_SmartPark_Team6.pdf

ORIGINALITY REPORT

8%	6%	6%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	github.com Internet Source	2%
2	Singireddy, Kalyan Srinivas Reddy. "Active Safety System Model to Avoid Rear-End Vehicle Collision Using Controlled Area Network (CAN).", Texas A&M University - Kingsville, 2020 Publication	2%
3	www.coursehero.com Internet Source	1%
4	impressions.manipal.edu Internet Source	1%
5	avrchip.com Internet Source	<1%
6	www.researchgate.net Internet Source	<1%
7	community.arm.com Internet Source	<1%
8	scholarscompass.vcu.edu Internet Source	

		<1%
9	Fuad Aliew. "An Approach for Precise Distance Measuring Using Ultrasonic Sensors", IECMA 2022, 2022 Publication	<1%
10	www.adb.org Internet Source	<1%

Exclude quotes	On	Exclude matches	< 3 words
Exclude bibliography	On		