S.D.M. COLLEGE OF ENGINEERING AND TECHNOLOGY, DHARWAD–580002

(An autonomous Institute affiliated to Visvesvaraya Technological University, Belagavi-590016)

Department of Electronic and communication Engineering

Technical Report

“Smart Park Assist”

Prepared By

ASHWATH H - 2SD24EC021

Student of **4th** semester

Under the guidance

Dr. Siddalingesh S. Navalgund

Assistant Professor

Dept. of ECE, SDMCET Dharwad-02

S.D.M. COLLEGE OF ENGINEERING AND TECHNOLOGY, DHARWAD–580002

CERTIFICATE

This is to certify that ASHWATH H, 2SD24EC021 has successfully designed and developed the project titled "SmartPark Assist: An IoT-Based Parking System" during the academic year 2025-26. This independent project demonstrates technical proficiency in IoT Architecture, Embedded C Programming, and Hardware Integration using the ESP32 platform. The work was carried out as a self-directed initiative to build a functional prototype for real-world automotive safety. This certificate recognizes the innovation, practical engineering logic, and dedication applied to the successful execution of this project.

Project Guide:	**Head of department:**
Dr. Siddalingesh S. Navalgund	**Dr. Shreedhar A Joshi**

Date: 27/02/ 2026

Place: SDMCET, Dharwad

1.Introduction

Parking in tight or crowded spaces is a constant challenge for drivers, often leading to avoidable accidents and vehicle damage. While cars built-in sensors, many drivers still struggle with “Blind spots” and poor distance judgement. Smart Park Assist offers a smart, low cost solution by using ultrasonic technology to “see” obstacles in real time.

By using the combination of Esp 32 microcontroller and Sound wave sensors, the system warns the driver in two way: through a light indication and live update in their phone. This project makes parking safer, easier and much more accurate by a tiny computer into a personal parking assistant.

2.Aim

To design and implement an intelligent parking assistance system that provides both visual (LED) and digital (Web Interface) real-time feedback to prevent vehicle collisions during parking.

3. Objective

a. High precision Distance Measurement:

To utilize ultrasonic sound wave propagation to detect obstacles with a resolution of 1cm. By calculating the “Time of Flight” (The time taken for the sound pulse to bounce back), the system provides an accurate digital measurement of the remaining space.

b. Visual warning system:

To design a 5-stage LED bar graph that acts as a physical proximity gauge. This provides the driver with an accurate status:

- Green(safe); Distance >10cm
- Red (Critical): Distance<10cm

This mimics the parking sensors found in high-end luxury vehicles.

c. Standalone IoT Integration:

To configure the ESP32 as a Wireless Access Point. This objective ensures the system functions as a "Private IoT Cloud," allowing any smartphone to connect to the "ESP32" Wi-Fi network to view a live, high-definition distance dashboard without needing an internet connection or a router.

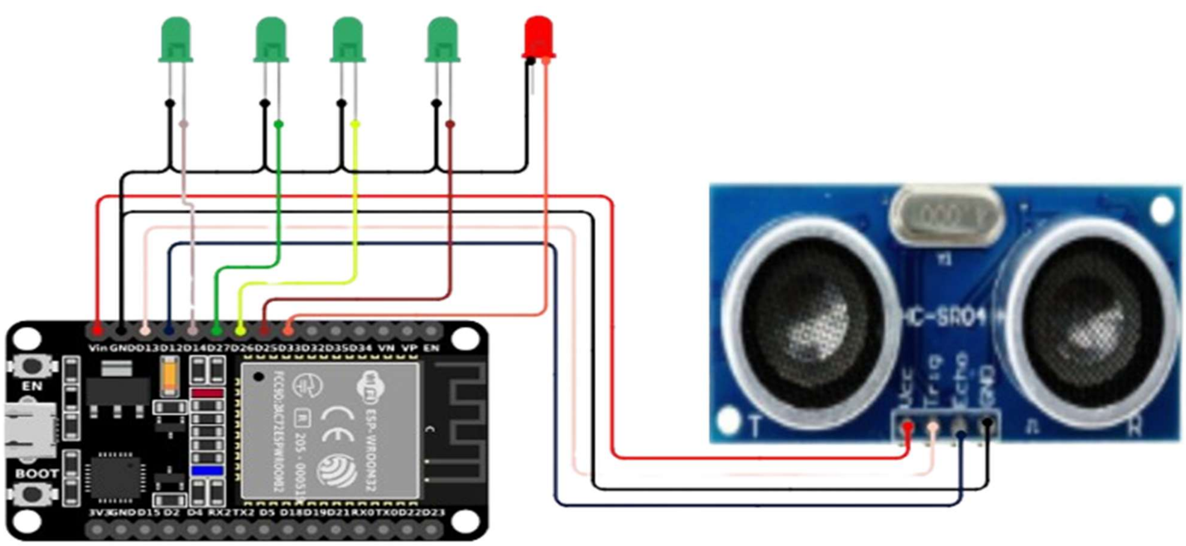
d. Real-Time Latency Optimization

To maintain a system refresh rate of 10Hz (100ms). This ensures that the distance displayed on the web interface and the LED status are synchronized with the vehicle's movement in real-time, preventing "lag-induced" collisions.

4.Components required

Component	Specification	Function	Quantity
Esp 32 microcontroller	Dual-core 240MHz	System processing and Wi-Fi hosting	1
HC-SR04 Sensor	2-400cm Range	Ultrasonic distance measuring	1
LED	Green, Red	Physical proximity bar graph	Green (4), Red (1)
Power supply	5V DC/USB	To power the system	1

5.Block diagram



6. Working Principle

The Science of "Echo-Location"

The system operates on the Time-of-Flight (ToF) principle, a method used by both advanced radar systems and nature (bats and dolphins) to map surroundings using sound. It translates physical sound waves into digital data through a four-stage process:

a. Signal Emission (The "Ping")

The process begins when the ESP32 sends a precise 10-microsecond high pulse to the Trig (Trigger) pin of the ultrasonic sensor. This command tells the sensor to emit eight short bursts of ultrasonic sound at 40 kHz—a frequency that is silent to human ears but perfect for bouncing off solid surfaces.

b. Echo Reception (The "Return")

Once the sound waves hit an obstacle (like a wall or a car bumper), they bounce back toward the sensor. The Echo pin on the sensor goes "HIGH" the moment the sound is sent and stays high until the reflected sound is received back. The ESP32 measures exactly how long (in microseconds) that Echo pin remained active.

c. The Mathematical Conversion

To turn that "time" into a readable "distance," the ESP32 uses the speed of sound, which is approximately 343 meters per second (or 0.0343 cm/μs). The formula is:

$$Distance=(Time*0.03430)/2$$

Why we divide by 2: The sound pulse travels a round trip—from the sensor to the object and back again. We divide the total duration by two to calculate the one-way distance between the sensor and the obstacle.

d. Data Logic & Multi-Platform Alerting

Once the distance is calculated, the "Brain" (ESP32) performs two tasks simultaneously:

- The Physical Alert: It checks the distance against set thresholds. If the distance is less than 50cm, it begins lighting up the LEDs. If it drops below 10cm, it triggers the Red LED (LED 5) as a critical stop signal.
- The Digital Alert: The ESP32 updates its internal Web Server. Any device connected to its private Wi-Fi hub receives this new distance value every

- 100 milliseconds, providing a live, high-speed digital readout on the smartphone dashboard.

7. How It Works (System Setup)

This section explains how the system talks to your phone and manages the warning lights.

a. Your Own Private Wi-Fi

The ESP32 creates its own Wi-Fi network called "ESP32". You don't need a router or the internet. Just connect your phone, and a live "Parking Dashboard" pops up in your browser. It updates twice every second so the numbers move as your car moves.

b. The LED "Warning Bar"

The 5 LEDs act like a progress bar for safety. As you get closer, the lights "climb" higher:

- Level 1 (50cm): First light on — Start paying attention.
- Level 2 (40cm): Second light on — Slow down.
- Level 3 (30cm): Third light on — Getting close.
- Level 4 (20cm): Fourth light on — Very close.
- Level 5 (10cm): RED ALERT — Stop immediately!

c. The Loop

The system repeats this check every 100 milliseconds. It is constantly "pinging" the wall, updating the lights, and sending the exact distance to your phone at the same time.

8. User Interface & Controls

This section covers how the driver actually interacts with the system.

* Wireless Dashboard: A clean, high-contrast screen that opens on any smartphone browser.

* Live Distance Meter: Large, bold numbers that show the remaining gap in centimeters, making it easy to read at a glance.

* Visual Progress Bar: An on-screen bar that changes colour (Green to Red) to match the physical LEDs on the device.

* One-Touch Connection: No apps to download; users simply connect to the Wi-Fi and the data starts flowing immediately.

* System Status: A "Heartbeat" indicator on the screen that confirms the sensor is active and the connection is stable.

9. Results & Observations

Now we look at how the system performed during actual use:

- Pinpoint Accuracy: The sensor successfully detected objects as small as a pole and as large as a wall with 1cm precision.
- Instant Response: There was no noticeable delay; as the car moved, the 10Hz (100 milliseconds) refresh rate kept the lights and screen perfectly synced.
- Reliable Range: The system worked perfectly from 2cm all the way to 400cm, covering the entire parking process.
- Connection Stability: The private Wi-Fi remained strong even through car windows and from several meters away.

10. Applications (Real-World Use)

This system is more than just a garage helper; it has several practical uses:

- Home Garage Safety: Helps drivers park perfectly in small spaces without hitting shelves or walls.
- Commercial Parking Lots: Can be installed in every slot to tell a central system which spaces are empty or full.
- Industrial Warehousing: Mounted on forklifts or storage racks to prevent collisions during heavy lifting.
- Tank Level Monitoring: The same logic can be used to measure the water or fuel level in a tank without touching the liquid.
- Smart Blind-Spot Detection: Can be used on trailers or large trucks to alert drivers of hidden obstacles during low-speed turns.

11. Future Usage

a. AI-Powered Object Detection

Instead of just measuring distance, we can add a camera module (ESP32-CAM) to the system. Using Artificial Intelligence, the device will not only know how far an object is but also what it is—distinguishing between a wall, another car, or even a person walking behind the vehicle for much smarter alerts.

b. Active Emergency Braking

We can move from "warning" the driver to "protecting" the car. By connecting the ESP32 directly to a vehicle's braking system (or a robot's motor), the software can be programmed to automatically apply the brakes if the driver enters the "Red Zone" too fast, physically preventing a collision before it happens.

12.Conclusion

The Smartpark Assist successfully demonstrates how IoT technology can solve everyday problems like parking safety. By combining high-precision ultrasonic sensors with the wireless power of the ESP32, we created a system that is both reliable and easy to use.

The project achieved its goal of providing dual-layer protection: physical LED feedback for a quick glance and a wireless dashboard for exact precision. This low-cost solution proves that smart technology can effectively eliminate "blind spots" and prevent vehicle damage, making parking safer for everyone.

13.Reference

a. Espressif Systems, "ESP32 Series Datasheet and Technical Reference Manual". https://www.espressif.com. (Referenced for configuring the Wi-Fi SoftAP, GPIO pin mapping and connections).

b. HC-SR04 User Manual, "Ultrasonic Sensor Module: Technical Specifications and Timing Diagrams". (Used for the distance calculation logic: Distance = (Time \times 0.0343) / 2).

c. Arduino, "Arduino IDE Language Reference and ESP32 Core Library Documentation". https://www.arduino.cc. (Used for developing the firmware, handling the WiFi.h library, and flashing the code).