iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79907

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

IoT-Based Smart Parking System Using Time-of-Flight Sensors with ANPR-Assisted Fault Detection

Sujal Khant[1], Ridhdhi Naik (guide)[2]
[1]*UG Final year student, Asha M. Tarsadia Institute of Computer Science and Technology, Uka Tarsadia University, India,*
[2]*Assistant Professor, Asha M. Tarsadia Institute of Computer Science and Technology, Uka Tarsadia University, India*

Abstract: ***The number of vehicles in crowded cities is increasing rapidly, making it a daily challenge for drivers to find a parking space. Instead of going directly to a free spot, drivers often keep moving around the parking area until they find one, which wastes fuel, adds to congestion, and increases carbon emissions. Although many IoT-based smart parking systems have been proposed in recent years, most of them rely on single-sensor setups and do not provide a reliable way to check whether a sensor is working properly or has failed. In this paper, we present a smart parking system built on a two-tier IoT architecture to improve reliability. VL53L0X Time-of-Flight sensors, connected with ESP32 microcontrollers, are used to detect vehicle presence in each parking slot and send the data through MQTT to a central Raspberry Pi. At the same time, the Raspberry Pi runs an OpenCV-based Automatic Number Plate Recognition (ANPR) module using a CCTV camera feed. This camera setup performs two tasks: it monitors vehicles entering and exiting the parking area, and it also helps verify sensor data to identify faulty or stuck sensors. A mobile application is provided to show drivers real-time availability of parking slots. After reviewing nine recent research papers in this domain, we observed that none of them used ToF sensors for slot detection, and none included a proper hardware fault-detection mechanism. The proposed system attempts to address both of these limitations.***
Keywords: ANPR, ESP32, Flutter, Internet of Things, MQTT, OpenCV, Raspberry Pi, Time-of-Flight Sensor

I. INTRODUCTION

Finding a parking space in busy urban places such as hospitals or shopping centers has become a common problem for drivers. When vehicles enter a crowded parking area, drivers usually have to go around different lanes again and again before they are able to find an empty parking space. Because of this, they spend more time driving inside the parking area. A study by Fahim et al. [1] reports that nearly 30% of vehicles in busy city areas are searching for parking at any given time. Khanna and Anand [2] also showed that using IoT-based systems in parking management can help reduce this wasted time.

Vehicle ownership in rapidly growing cities continues to increase every year, but parking infrastructure often does not grow at the same pace. Many parking facilities still rely on manual management, where a staff member at the entrance distributes paper tickets to drivers. In such systems, the staff can only estimate whether parking spaces are available. Drivers entering the parking area usually have no clear information about where empty slots are located, so they must search for a space themselves. This situation leads to inefficient use of parking capacity and increases unnecessary vehicle movement within the facility. As urban areas continue to expand, there is a growing need for more organized and technology-based parking management solutions. IoT-based systems make this possible by allowing real-time monitoring and communication between devices and users [2, 14].

In this work, we propose a smart parking architecture designed to improve the reliability and efficiency of parking management. The system uses VL53L0X Time-of-Flight (ToF) sensors to detect vehicles at individual parking slots. These sensors provide more accurate distance measurements compared to commonly used ultrasonic or infrared sensors. Each parking slot is connected to an ESP32 node that collects sensor data and sends it to a central Raspberry Pi controller. The Raspberry Pi gathers and processes this information and makes it available through programmable interfaces. In addition, an OpenCV-based Automatic Number Plate Recognition (ANPR) system is placed at the entry and exit points of the parking area. This system counts vehicles entering and leaving the facility and compares the results with the sensor data. If any difference is detected, the system can identify possible sensor errors and improve the overall reliability of the parking management process.

II. LITERATURE REVIEW

We reviewed nine research papers on IoT-based smart parking systems published between 2018 and 2024. The main goal of this review was to study the hardware setups used in recent smart parking solutions and to identify common limitations in their designs. Singh et al. [3] proposed an IoT-based parking sensor network designed for a smart campus environment.

©IJRASET: All Rights are Reserved

In their system, HC-SR04 ultrasonic sensors are connected to a NodeMCU microcontroller to detect the occupancy of parking spaces. The sensor data is transmitted through Wi-Fi to a cloud database using Google Firebase, and a mobile application developed with MIT App Inventor 2 displays parking availability to users. A similar system in [4] used a Raspberry Pi together with ultrasonic sensors and a Pi Camera module. The system also included an online payment feature that allows users to reserve parking spaces and pay through a mobile application using PayPal. In this design, both the ultrasonic sensors and the camera are used to detect the presence of vehicles and to monitor the parking area. However, the system does not implement a mechanism to cross-verify the sensor readings using camera data. This observation motivated the use of a camera module in our proposed system as an additional verification mechanism to detect vehicle presence and improve the reliability of parking slot detection.

Muzhafar et al. [7] proposed an IoT-based parking system that uses infrared (IR) sensors connected to a NodeMCU to detect whether a parking slot is occupied. The system checks the parking spaces and shows the number of available slots on an LCD screen placed at the parking entrance. The sensor data is also saved in a database to keep track of vehicle entry and exit times. This system offers a simple and low-cost way to monitor parking availability. However, the parking information is only shown on the LCD screen at the entrance, and the system does not provide a mobile application or remote access for drivers before they arrive at the parking area. The study in [12] presents a smart parking system that uses both infrared and ultrasonic sensors controlled by an Arduino Mega 2560. The system also includes an RFID reader to allow vehicles to enter the parking area. An LCD screen at the entrance shows parking information, and a mobile application connected through Wi-Fi allows users to check available parking spaces.

Another system described in [5] uses an ESP8266 NodeMCU with ultrasonic sensors to detect whether a parking slot is occupied. In this design, users can scan a printed QR code to open a web dashboard where they can see the parking status. This allows drivers to check parking availability through a simple online interface.

Agnihotri et al. [9] proposed a different approach using RFID technology to identify vehicles entering a parking area. In their system, each vehicle must carry an RFID tag so that it can be recognized by the reader at the entrance. The system records vehicle information and manages parking access through a web application. Although this method provides accurate vehicle identification, it requires every vehicle to have an RFID tag, which may not be practical for large public parking areas.

Several studies have focused on using cloud platforms to monitor parking systems. Abdulsaheb et al. [8] developed a system that uses infrared (IR) sensors and sends the parking data to a cloud platform so that it can be monitored remotely. Shahu et al. [6] proposed a system that combines Arduino and NodeMCU boards to collect ultrasonic sensor data and upload the parking status to the ThingSpeak cloud platform.

A more complex system was presented by Mamun et al. [13]. Their design uses multiple sensors, including IR sensors to detect vehicles in parking slots, a DHT22 sensor to measure temperature and humidity, and an MQ-2 gas sensor to monitor air quality. These sensors are connected through MQTT communication using both Arduino and Raspberry Pi boards, and the system also includes a mobile application for users.

However, the authors mentioned that IR sensors can sometimes lose accuracy in real environments, and their system did not include a backup method to verify the sensor readings.

Fahim et al. [1] studied many smart parking systems used in different parts of the world. In their review, they examined several technologies used to detect whether a parking space is occupied, including ultrasonic sensors, magnetic sensors, infrared sensors, and camera-based systems. The authors discussed how these sensing methods work and compared their advantages and disadvantages in different parking situations. Their study shows that many existing parking systems rely on only one type of sensor to detect vehicles. Because of this, the system's performance depends heavily on the accuracy of that sensor. If the sensor gives incorrect readings, the system may show wrong information about available parking spaces.

From the review of the existing studies, two common limitations can be observed. First, none of the reviewed systems use Time-of-Flight (ToF) sensors for parking slot detection. Second, the reviewed systems do not include a secondary mechanism to verify sensor readings or detect possible sensor failures.

Table 1 summarizes the main features of the existing systems, including the sensors used, processing hardware, application support, and whether any fault detection or ANPR mechanism is included.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

TABLE I
Comparison of Existing Smart Parking Implementations

Ref.	Controller	Sensors	Communication	User Interface	Fault Detection	ToF Sensor Used
[3]	NodeMCU	HC-SR04 Ultrasonic	Firebase	Mobile App	No	No
[4]	Raspberry Pi	Ultrasonic + Camera	Local DB	Mobile App	No	No
[5]	ESP8266 Node MCU	Ultrasonic Sensor	Local DB	QR + Web Dashboard	No	No
[6]	Arduino + NodeMCU	Ultrasonic Sensor	ThingSpeak	User application	No	No
[7]	NodeMCU	IR Sensors	Local DB	LCD Only	No	No
[8]	Node MCU	IR Sensors	Cloud	Remote Monitor	No	No
[9]	-	RFID	Web App	Web App	No	No
[12]	Arduino Mega 2560	IR + Ultrasonic + RFID	Local DB	Mobile App + LCD	No	No
[13]	Arduino + RPi	IR + DHT22 + MQ-2	MQTT + Local DB	Mobile App + OLED screen	No	No
Proposed	ESP32 + RPi 4	VL53L0X ToF	MQTT + REST	Mobile App	Yes	Yes

III. PROPOSED ARCHITECTURE

A. System Overview

The architecture of the proposed system is shown in Fig. 1. The system has two main levels. The first level uses ESP32 microcontroller nodes to manage parking slots in a parking zone. Each slot is monitored by a VL53L0X Time-of-Flight (TOF) sensor mounted on the ceiling. The ESP32 reads the distance values from the sensor through an I2C connection and checks whether the slot is occupied. It then sends the parking status as an MQTT message through the local WiFi network. The second level is a central Raspberry Pi server placed in the control room. This server runs an MQTT broker to receive updates from the ESP32 nodes and also provides a REST API for the mobile application. In addition, the Raspberry Pi processes the camera feed to perform Automatic Number Plate Recognition (ANPR) using OpenCV and Tesseract OCR.

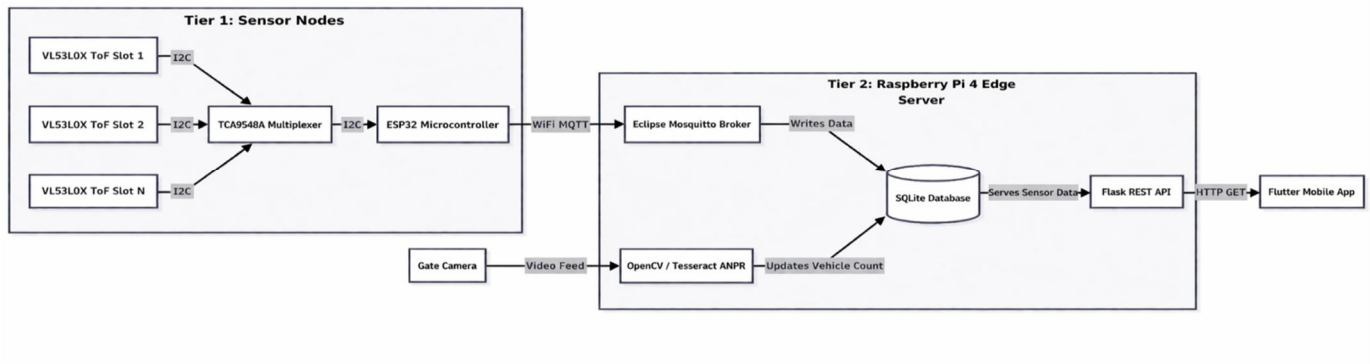


Fig. 1 Two-tier architecture of the proposed parking system.

MQTT was chosen instead of the traditional HTTP protocol because it works better for IoT systems. It is lightweight and suitable for sending small messages between devices. Jaloudi [14] discussed the use of MQTT in smart city applications where many devices send small data packets frequently. The publish–subscribe communication model also makes the system easy to expand. New ESP32 sensor zones can be added without making major changes to the central server.

B. Sensor Selection and Node Configuration

A VL53L0X Time-of-Flight (ToF) sensor measures distance by sending a small infrared laser pulse and calculating how long it takes for the light to return after hitting an object. Based on this time measurement, the sensor calculates the distance between the sensor and the object. In the proposed system, the sensor is mounted on the ceiling above each parking slot. When the slot is empty, the reflected signal usually comes from the ground, which results in a larger measured distance. When a vehicle is parked in the slot, the laser reflects from the roof of the car, which produces a shorter distance value. The system compares the measured distance with a predefined threshold to determine whether the parking slot is occupied or free.

Why is the proposed setup better than traditional sensors? Ultrasonic sensors measure distance using sound waves, and their accuracy can be affected by environmental conditions because the speed of sound changes with temperature [17]. In very hot weather, this change can influence the measured distance and lead to inaccurate readings. In contrast, the VL53L0X sensor measures distance using laser-based Time-of-Flight (ToF) technology, which calculates the travel time of a light pulse to determine distance [15].

Traditional infrared (IR) proximity sensors can also face problems in outdoor environments because sunlight contains strong infrared radiation that may interfere with the sensor signal [16]. Time-of-Flight sensors use a focused laser signal for ranging, which makes them more stable under different lighting conditions. Because of this, ToF sensors can provide more reliable distance measurements for outdoor parking applications [10].

In the proposed architecture, one ESP32 board is used to manage multiple sensors in order to reduce the overall system cost. The VL53L0X Time-of-Flight sensors communicate using the I2C interface and share the same default address. Because of this, multiple identical sensors cannot be connected directly to the same bus without causing address conflicts. To solve this problem, a TCA9548A I2C multiplexer is used, which allows a single microcontroller to communicate with several sensors that have identical addresses [15]. Using this approach, the ESP32 reads the distance values from each sensor sequentially and determines the status of every parking slot. After collecting the data from all sensors in the zone, the ESP32 sends the information wirelessly to the central server for further processing.

C. Edge Server Processing

The Raspberry Pi acts as the central controller of the system. It runs an Eclipse Mosquitto MQTT broker that receives messages from the ESP32 sensor nodes. A Python program runs in the background to process these messages. It stores the parking slot status in a local SQLite database and provides the data to the mobile application through a REST API.

At the same time, the Raspberry Pi processes video from a camera placed at the parking entrance and exit. The ANPR software first converts the video frames to grayscale and then uses edge detection to find possible license plate regions. After locating the plate area, the image is passed to the Tesseract OCR engine to read the characters on the plate [11]. When a vehicle enters the parking area, its plate number and time are recorded. When the vehicle leaves, the system finds the matching plate number and removes it from the active list. This process helps the system keep an accurate count of the vehicles currently inside the parking facility.

D. Fault Detection Mechanism

One thing that makes this system different from most others is that it does not fully trust the sensors on their own. In most smart parking systems, whatever the sensor says is taken as fact [1, 13]. But in a real parking lot, sensors can start giving wrong readings for simple reasons like dust building up on the lens, a loose wire, or a small power issue. When that happens, the system keeps running normally, showing wrong slot availability to drivers with no indication that anything is wrong.

To get around this, the proposed system checks two separate sources of data against each other. The ToF sensors report how many slots are showing as occupied based on measured distance. At the same time, the ANPR camera keeps a running count of how many vehicles have entered the lot and not yet left. These two numbers should always be close to each other. If the sensors say 50 slots are occupied but the camera count shows 53 vehicles inside, something does not add up. A gap like that is a sign that one or more sensors may have stopped working properly and need attention.

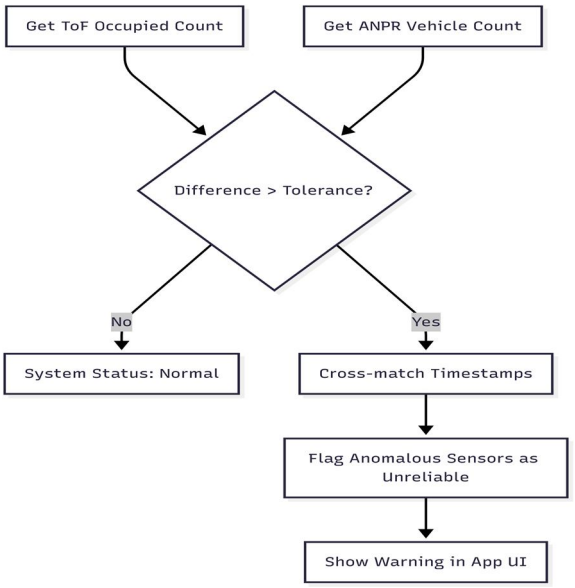


Fig. 2 Flowchart showing sensor fault detection through ANPR cross-validation.

E. User Application Interface

The system also includes a mobile application developed using Flutter. The app communicates with the Raspberry Pi server by sending HTTP GET requests to the REST API to obtain the latest parking slot information. The application shows a visual layout of the parking area so that users can easily understand the current status of each slot. In the interface, a green color indicates that a slot is available, while a red color shows that the slot is occupied. If the system detects that a sensor may not be working properly, that slot is shown in yellow. This helps drivers avoid relying on incorrect information and improves the overall reliability of the parking system.

TABLE II
Software Stack

Software / Layer	Implementation	Purpose
Python 3 + OpenCV	Image Processing Pipeline	Image processing and contour detection
Tesseract OCR	OCR Engine	Extracting alphanumeric text from plates
Eclipse Mosquitto	MQTT Broker	Lightweight MQTT broker
Flask (Python)	REST API Server	REST API for the mobile application
Flutter	Mobile Application	Cross-platform mobile front-end

TABLE III
Hardware Components

Component	Specification	Primary Role
VL53L0X	Up to 2 m (1.2 m default mode)	Capturing occupancy
ESP32	Dual-core, 240 MHz, Wi-Fi	Operating sensor nodes
TCA9548A	8-channel I2C mux	Managing multiple ToFs
Raspberry Pi 4	4 GB RAM, Wi-Fi	Acting as edge server
Camera Module	CSI, 1080p	Capturing plates for ANPR

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

IV. COMPARATIVE ANALYSIS

The proposed system offers several improvements over the smart parking systems found in the existing literature. One key difference is the use of Time-of-Flight (ToF) sensors instead of ultrasonic or infrared sensors. ToF sensors measure distance by sending a small laser pulse and calculating how long it takes for the light to return [10], which gives stable and accurate readings. Ultrasonic sensors rely on sound waves, and their readings can shift when the temperature changes because the speed of sound is affected by temperature [17]. Infrared sensors can also run into problems outdoors because natural sunlight carries infrared radiation that can interfere with the sensor signal [16]. Because ToF sensors avoid both of these issues, they are a more reliable choice for open or semi-covered parking environments.

Another advantage is the flexible two-tier architecture. ESP32 nodes collect data from the parking slot sensors and pass it to a central Raspberry Pi server. This structure makes it straightforward to expand the system. If the parking area grows and more slots are added, new ESP32 nodes can be installed and connected to the existing network without changes to the central server. Many earlier smart parking designs route everything through a single controller, which makes scaling the system harder.

Reliability is another important improvement. Most existing smart parking systems depend entirely on sensor data to decide whether a slot is free or occupied [1]. If a sensor fails or a connection is lost, the system continues to send wrong information to drivers with no way of knowing there is a problem. The proposed system reduces this risk by using an ANPR camera as a secondary data source. The camera keeps a count of vehicles that have entered and not yet left the parking area. When this count is compared against the number of slots the sensors report as occupied, any large difference points to a possible sensor fault, which the system can then flag for the driver interface.

V. CONCLUSION

This paper described the design of an IoT-based smart parking system that tries to go beyond what most existing systems currently offer. The two main problems it targets are inaccurate slot detection caused by sensor limitations and the complete lack of fault detection in current designs.

The proposed system replaces ultrasonic and infrared sensors with VL53L0X Time-of-Flight sensors. These sensors are not thrown off by temperature changes in the air or bright outdoor light, which makes them a better match for real parking environments. The two-tier setup, with ESP32 nodes at each zone feeding into a central Raspberry Pi, keeps the system practical and easy to grow as a parking facility expands.

What sets this work apart from the nine systems reviewed is the use of an ANPR camera not just for identifying vehicles, but as a way to check whether the sensors are still working correctly. By comparing the camera's vehicle count with the sensor occupancy data, the system can catch faults that would otherwise go unnoticed and make sure drivers are not sent to a slot that is already taken.

A. Future Work

This paper covers the design and architecture only. The next step is to build a working prototype and test it in an actual parking environment. Testing will help measure things like MQTT message delay and how well the ANPR module reads plates under different lighting conditions. There is also potential to use past parking data with machine learning to give drivers a prediction of when slots are likely to be free.

REFERENCES

[1] A. Fahim, M. Hasan, and M. A. Chowdhury, "Smart Parking Systems: Comprehensive Review Based on Various Aspects," Heliyon, vol. 7, no. 5, p. e07050, 2021. [Online]. Available: https://doi.org/10.1016/j.heliyon.2021.e07050

[2] A. Khanna and R. Anand, "IoT based smart parking system," in Proc. Int. Conf. Internet of Things and Applications (IOTA), IEEE, 2016, pp. 266–270. [Online]. Available: https://doi.org/10.1109/IOTA.2016.7562735

[3] J. J. Singh, N. N. Ravi, and P. S. Krishnan, "IoT based Parking Sensor Network for Smart Campus," Int. J. Eng. Technol., vol. 7, no. 4.35, pp. 26–34, 2018.

[4] M. M. S. Ismail, M. Jusoh, T. Sabapathy, M. N. Osman, H. A. Rahim, M. N. M. Yasin, and A. F. M. Fazilah, "IoT Based Smart Parking System," J. Phys.: Conf. Ser., vol. 1424, no. 1, p. 012021, 2019. [Online]. Available: https://doi.org/10.1088/1742-6596/1424/1/012021

[5] S. Y. C. Hong, C. C. Kang, J. D. Tan, and M. Ariannejad, "Smart Parking System Using IoT Sensors," J. Eng. Technol. Appl. Phys., vol. 5, no. 1, pp. 63–69, 2023. [Online]. Available: https://doi.org/10.33093/jetap.2023.5.1.7

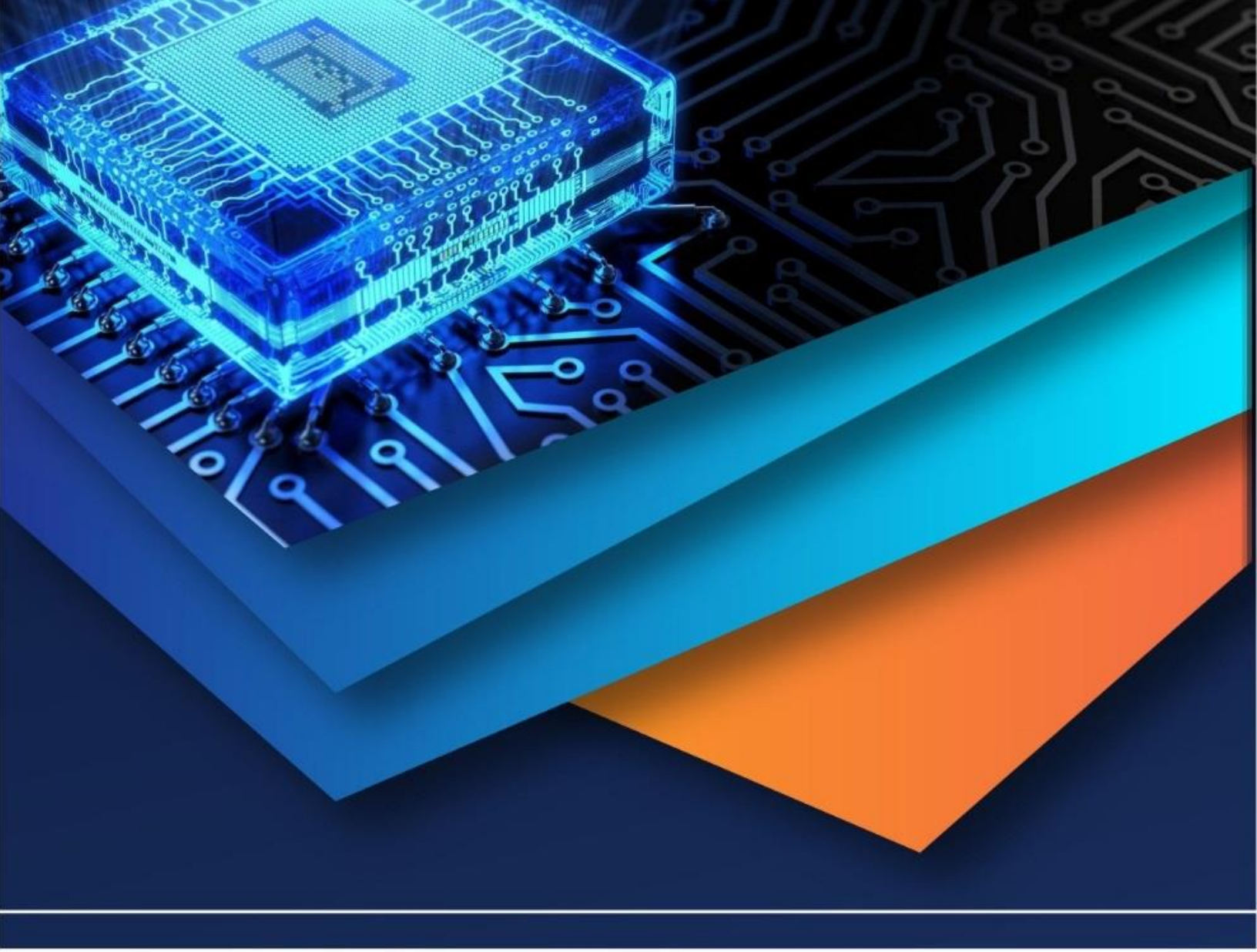
[6] V. Shahu, S. Padole, S. Mahajan, P. Tiple, R. C. Iyer, and V. G. Girhepunje, "Smart Parking Slot Management System Using IoT," Int. J. Sci. Res. Eng. Manag. (IJSREM), 2024. [Online]. Available: https://doi.org/10.55041/IJSREM29994

[7] M. A. K. Muzhafar, M. N. Osman, N. A. Othman, and K. A. Sedek, "IoT based Car Parking Management System using IR Sensor," J. Comput. Res. Innov., vol. 5, no. 2, pp. 75–84, 2020. [Online]. Available: https://doi.org/10.24191/jcrinn.v5i2.151

[8] J. A. Abdulsaheb, R. M. Nafea, W. A. M. Al-Jawher, and M. L. Hayyawi, "IoT Based Smart Parking System," J. Port Sci. Res., vol. 7, no. 3, p. 1, 2024. [Online]. Available: https://doi.org/10.36371/port.2024.3.1

[9] S. Agnihotri, A. Kumar, G. Singh, P. Dixit, S. Ahmad, and S. Chaurasia, "IOT Based Smart Parking System," Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET), vol. 12, no. V, 2024. [Online]. Available: https://doi.org/10.22214/ijraset.2024.61888
[10] G. C. Sanchez and J. A. Chavez, "A Low-Cost Occupancy Control Solution Using a Time-of-Flight Ranging Sensor Laser," CEUR Workshop Proceedings, 2022. [Online]. Available: https://ceur-ws.org/Vol-3059/
[11] K. Kumawat, A. Jain, and N. Tiwari, "Relevance of Automatic Number Plate Recognition Systems in Vehicle Theft Detection," Engineering Proceedings, vol. 59, no. 1, p. 185, Jan. 2024. [Online]. Available: https://doi.org/10.3390/engproc2023059185
[12] Y. Allbadi, J. N. Shehab, and M. M. Jasim, "The Smart Parking System Using Ultrasonic Control Sensors," IOP Conf. Ser.: Mater. Sci. Eng., vol. 1076, no. 1, p. 012064, 2021. [Online]. Available: https://doi.org/10.1088/1757-899X/1076/1/012064
[13] A. A. Mamun, A. Hasib, A. S. M. Mussa, R. Hossen, and A. Rahman, "IoT-Enabled Smart Car Parking System through Integrated Sensors and Mobile Applications," arXiv preprint arXiv:2412.10774, 2024. [Online]. Available: https://arxiv.org/abs/2412.10774
[14] S. Jaloudi, "MQTT for IoT-based Applications in Smart Cities," Palestinian Journal of Technology and Applied Sciences (PJTAS), no. 2, pp. 1–13, 2019. [Online]. Available: https://journals.qou.edu/index.php/PJTAS/article/view/1550
[15] STMicroelectronics, "VL53L0X Time-of-Flight Ranging Sensor Datasheet," STMicroelectronics, 2020. [Online]. Available: https://www.st.com/resource/en/datasheet/vl53l0x.pdf
[16] P. Ajmera, "A Review Paper on Infrared Sensor," Int. J. Eng. Res. Technol. (IJERT), vol. 5, no. 23, 2018. [Online]. Available: https://www.ijert.org/a-review-paper-on-infrared-sensor
[17] A. Carullo and M. Parvis, "An Ultrasonic Sensor for Distance Measurement in Automotive Applications," IEEE Sensors Journal, vol. 1, no. 2, pp. 143–147, Aug. 2001. doi: 10.1109/JSEN.2001.936931.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)