IJARSCT
IJARSCT
ISSN: 2581-9429
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Volume 6, Issue 1, May 2026
Impact Factor: 8.2

Smart Parking System for Real-Time Slot Detection

Shruti Saxena, Anshima Manu, Harsh Tiwari
Department of Computer Science and Engineering (IoT)
Raj Kumar Goel Institute of Technology, Ghaziabad, India
shrutfio@rkgit.edu.in, anshimamanu@gmail.com, haarstiiwari@gmail.com

Abstract: *The current paper is an implementation and design of a smart parking management system based on an ESP32 microcontroller, HC-SR04 ultrasonic sensors, and the utilization of the Blynk IoT platform. The system proposed is used to track the occupancy of individual parking slots in real time through taking the distance of the objects (vehicles) at the ultrasonic level. Slot status (Vacant, Occupied) is wirelessly sent to the Blynk cloud dashboard through Wi-Fi and displayed as a slot along with its status label. The system was also designed with a virtual switch on the Blynk dashboard, which kept only vacant slots visible on the dashboard except when the virtual switch is activated so that drivers can easily locate available parking lots. The test performance shows that it can be effectively used and its cost is reasonable, as it has been experimentally proven that accurate detection was achieved within response times less than two seconds to effectively manage modern parking in the city.*

Keywords: Smart parking, ESP32, ultrasonic sensor, parking management, Blynk IoT,HC-SR04, real-time monitoring

I. INTRODUCTION

The rapid urbanization and city growth has resulted in a proliferation of cars on streets around the globe. One of the challenges resulting from this is finding parking spaces in congested cities. This problem leads to much time wasted by drivers looking for parking spaces, which increases traffic congestion, fuel consumption, and air pollution [1].

As the Internet of Things (IoT) grows, new tools are available to address urban challenges. The IoT is the interconnection of physical devices that exchange data and route information. For parking systems, IoT allows automatic monitoring of parking spaces and real-time feedback to the users and system administrators. It can be used to minimise human intervention and increase effectiveness of parking systems [2].Several additional large-scale IoT-based parking systems and optimization algorithms have also been proposed in other studies [9],[12].

The low-cost smart parking system presented in this paper is based on an ESP32 microcontroller that is connected to the internet using the built-in Wi-Fi. To determine whether the parking lot is free or occupied, the system relies on ultrasonic sensors (HC-SR04s) that emit a pulse of ultrasonic sound and measure the time it takes for the pulse to return from the object after being reflected.

Data from the sensor is transmitted to the cloud and presented on a user-friendly interface via the Blynk IoT application. The dashboard displays the available and occupied parking spaces in real-time on users' mobile devices or web browsers. This system also has a filter system that can display only the vacant parking spaces to make it convenient and quick way to find a parking space.

In addition to real-time monitoring, the system also supports basic occupancy prediction in a practical sense. The system provides real-time information to users about the occupancy status of each parking slot, which ismonitored and then displayed on a dashboard. This continuous mapping of parking availability helps users understand how parking spaces are currently being used. When users access this data as they navigate to or consider parking, it helps them make timely and informed decisions in real time. The continuous stream of live data and its user-friendly visualizationhelp in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, May 2026
Impact Factor: 8.2

practical prediction of occupancy, enabling users to assess the probability of finding parking spaces without wasting time searching. This not only improves user convenience but also promotes efficient parking space management.

The proposed system integrates IoT devices, wireless network communication and cloud monitoring to eliminate the need for manual monitoring and physical structures like signboards. It enhances user comfort and aids in driving decisions. These smart parking systems contribute to creating a smart city by enhancing the utilisation of parking resources and city transportation needs.

The remainder of this paper is structured as follows: Section II presents related work, Section III covers the architecture and hardware design of the system, Section IV is about the system implementation, Section V reports the experimental study and Section VI concludes this paper.

II. RELATED WORK

A number of parking systems based on the IoT has been suggested in the literature. An Arduino-based parking system with infrared sensors and LCD display was designed by Patil et al. [3], with a slot-level detection, but without a remote-accessibility feature. Acharya et al. [4] came up with a Raspberry Pi based vehicle detection image processing system; it was accurate but the hardware costs and computing overhead were huge.

Other applications of RFID technology that have been investigated are entry/exit management, however, they do not offer slot-level occupancy in real-time [5]. Optical sensing and hybrid sensing approaches have also been explored to improve detection reliability [10]. Most recent publications have included Wi-Fi-enabled microcontrollers like the NodeMCU or ESP8266 modules and MQTT brokers to publish sensor data to cloud dashboards, showing which can indicate the feasibility of Wi-Fi-enabled microcontrollers in smart parking.Other implementations, including those using NodeMCU and wireless sensor networks, have also highlighted the scalability and cost-effectiveness of parking [13], [15].

Its easy integration, virtual pin abstraction and cross platform support of mobile applications have been adopted in a few IoT prototyping works using the Blynk platform [7]. Cloud-based IoT architectures to support remote monitoring and controlling have also been widely studied [14]. Nevertheless, there are not many studies that combine ultrasonic sensing and Blynk systematically to create the scalable, multi slot parking display with a user-definable filtering. The gap is covered in the present work.

Moreover, the existing systems are also mostly focused on the accuracy of detection with no attention paid to user experience and interface simplicity. They are not very useful due to the absence of easy-to-use dashboards and real-time data-filtering options. Such cloud-based solution as Blynk can be used to determine the desired level of visualization and control flexibility and scalability. The proposed system, with its effective sensing with easy to navigate interface, will perhaps be appealing to fill the so-called gap between technical functionality and an easy design that is more suited by a real-world application.

III. SYSTEM ARCHITECTURE

Overview

The proposed system comprises three principal layers: (1) the sensing layer, consisting of HC-SR04 ultrasonic sensors mounted above each parking slot; (2) the processing layer, implemented on the ESP32 microcontroller; and (3) the presentation layer, hosted on the Blynk cloud dashboard accessible via the Blynk mobile application or web portal.These layered architectures are widely used for IoT based smart parking system due to their scalability and modularity [11].

The modularity and the scalability are another key property of the architecture. The system is also easily scalable to the addition of more sensors and software configurations as more parking slots are added to the system: each parking slot can add its sensor and readjust to the new settings of the configuration. The beauty of the Wi-Fi connectivity is that, it can transmit data through long distance without necessarily engaging other communication gadgets. The layered

DOI: 10.48175/IJARSCT-34725

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, May 2026
Impact Factor: 8.2

solution is easier to maintain the system and has more allowances to improve the system in future by adding mobile applications/ centralized servers/ smart city infrastructure to the system.

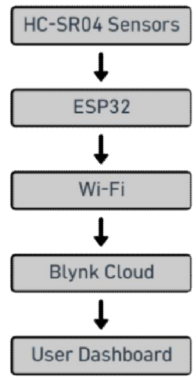

FIG I.: Hardware Components

The hardware is keyed components as outlined below.

ESP32 Microcontroller: ESP32 (Espressif Systems) is a 32-bit dual-core Xtensa LX6 that runs programs to up to 240 MHz It has stunned built-in 802.11 b/g/n Wi-Fi and Bluetooth, 520 KB SRAM and 34 programmable GPIO pins. It is a low-power and easily peripherally-supported chip that is specially adapted to the IoT [8].ESP32-based implementations have since been demonstrated for real-time sensing and communication in other smart environments [9].

HC-SR04 Ultrasonic Sensor: The working principle of HC-SR04 is based on the principle of the ultrasonic time-of-flight (ToF) measurement. It sends a burst of well above 40 kHz ultrasonic signal through the trigger pin and counts the intervals between the signal and the echo. Its effective range is 2 cm-400 cm that has a resolution of around 3 mm. The distance d is determined as:

$$d = (t \times v) / 2$$

where t is Echo pulse duration in micro seconds and v is speed of sound (around 343 m/s at 20 C). A distance is defined as a threshold to distinguish a slot as Occupied or Vacant.

Power: The system is being supplied by a 5V DC, which is supplied to the ESP32 through USB. Each HC-SR04 conducts about 15 mA; therefore, four sensors use about 60 mA, which is far under the sourcing capacity of the ESP32 when using an external power source on its GPIOs.

Connecting Wires: Electrically, a connection to the power supply, ultrasonic sensors, and ESP32 is done with connective wires. They guarantee adequate connection of signals and power across the system. Most flexible and more secure breadboard connections or module connections tend to use jumper wires (male-to-male and male-to-female). Wiring must be done properly to eliminate system reliability and signal loss or short circuits.

The hardware devices are determined by the cost, availability, affordability and performance reliability. ESP32 microcontroller has now become the most desired as it has learned the trick of providing two core processing unit and meshing on wireless communication without necessarily using other modules. The ultrasonic sensors also cost less and are more accurate as compared to the other sensing technologies. The hardware design offers, in general, a trade-off between efficiency and cheap cost, on small and large-scale set-ups of which the system is feasible.

TABLE I.: *Hardware Components Summary*

Component	Model	Quantity
Microcontroller	ESP32 DevKit V1	1
Ultrasonic Sensor	HC-SR04	4 (scalable)
IoT Platform	Blynk IoT	Cloud

Power Supply	5V USB / Adapter	1
Jumper Wires	M-M / M-F	~40

System Block Diagram

All HC-SR04 sensors are connected to a specific pair of ESP32 GPIO pins (Trigger and Echo). The ESP32 will be connected to the immediate Wi-Fi and will interact with the Blynk server via TCP/IP. The Blynk dashboard is set to display value widgets per slot and a virtual switch one to filter.

The continuous feedback between visualization and sensing which is indicated on the block diagram is the other factor. Data received on sensors is read and showcased on the dashboard in real time, this implies that it will update the users with data. Such live time synchronization is significant in making the system well maintained and the users are confident that it is so. Moreover, ordered data streams minimize processing delays as well as the notion of software hardware compatibility.

IV. SOFTWARE IMPLEMENTATION

Firmware Development

The ESP32 code is created in the Arduino IDE with the Blynk library (v1.3.2) and default ESP32 board support package. The operations of the main loop are the following: (1) acquire distance with each HC-SR04 sensor by calling the pulseIn() function; (2) compare the distance obtained with a specified threshold (15 cm in this example); (3) assign the state of the corresponding virtual pin either Occupied or Vacant status string; and (4) push the updates on the Blynk server conditionally by checking the state of the filter switch.

Software development also exists; stability and error-handling of the system. It has built-in the firmware and capabilities to manage sensor faults, faulty measurements, and network faults. When connected to online Wi-Fi network and Blynk server, ESP32 tries to connect automatically when there are some problems with the network. The features will guarantee full functionality of the systems and reduce system downtimes. The modular code design will also ensure that it can easily add and make future modifications in the code without the current codebase being significantly altered.

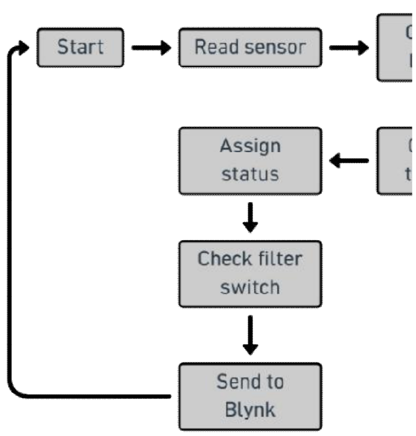

FIG II.: Blynk Dashboard Configuration

The Blynk 2.0 platform is set up with the following widgets allocated to virtual pins V1-V4 (one per slot): Label widgets that get filled with the existing status string and colour coding (green for Vacant, red for Occupied). Virtual pin V10 has a Button widget that is set in Switch mode to serve as the filter switch.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, May 2026
Impact Factor: 8.2

A switch widget (V10) is enabled: the firmware checks the status of each slot and only sends update strings to slots with a status of Vacant, but prevents or clears the display of slots with a status of Occupied. This gives them a filtered view, through which users can quickly see available spaces.

A dashboard will be user friendly, responsive and bare to ensure the users can easily understand the parking spot with no doubt. The layout of space occupied by the widgets is rational in order to present an easy view of the overall parking slots. The colour signals can be easily read and up to the minute information displays by keeping the information up to date appropriately. This is a basic design that enhances user interest and customization of the system to suit its everyday usage without the user necessarily having a technical knowhow in the same.

Filter Switch Logic

The pseudocode of the filter logic follows: When the filter switch (filterOn) is TRUE, the slots with status = Vacant will be written to their corresponding Blynk virtual pins, whereas the occupancy slots will be written as empty strings or a dash (= -) to represent suppression. When filterOn is not true, this will show all slots with their real status strings, giving a full picture of the parking lot.

The filtering system minimizes the usage particularly in big parking lots, in which the user can be bombarded by showing all the slots. The system helps them be less cluttered, as the system will not show areas that will not be available to users thus enabling them make decisions faster. We also worked on the logic to see that the switching states do not have much delay hence creating a smooth and responsive user experience in the dashboard.

Communication Protocol

The ESP32 connects to the Blynk cloud database via a connected TCP socket on port 80 (HTTP) or port 443 (HTTPS) (SSL). The Blynk library manages the connection, heartbeat and reconnection logic in a transparent manner. One of the data updates per slot is estimated around 1 Hz, so a four-slot system has a total bandwidth of around 8 updates per second, which is also far below the free-tier bandwidth of the Blynk platform.

Other considerations that need to be taken when conducting the communication exercise are security and reliability. The data exchanged between the ESP32 and the cloud server is encrypted with the help of the secure protocol like HTTPS. They also keep standard signals on heartbeat, so that the connection can run as well as to notice the events of disconnection. These will lead to the establishment of a stable system that is secure which is critical to the practical use of IoT.

V. EXPERIMENTAL RESULTS

Test Setup

The system was tested in a managed indoor parking place where there were four tagged slots. The sensors of HC-SR04 should be attached to the surface of the floor at a height of 25 cm, with the help of 3D-printed brackets. Test experiments also contained slots with scenarios in different combinations of occupied and vacant states. A standard passenger vehicle and a cardboard box (as a proxy vehicle) were used to simulate parked vehicles.

Detection Accuracy

Table II is a summary of the detection accuracy results with 100 trials at a time (400 trips at a time). The system had a total accuracy of detection of 98.5 percent with false positives mainly at the edges of slots in which the beams of sensors might overlap with other slots. A decrease in the threshold imposed to 12 cm instead of 15 reduced the boundaries interference and increased the accuracy to 99.25.

Response Time

Measurement covers 50 end-to-end trial times in the latency of vehicles placement/removal to dashboard status update. The mean time response was 1.34 seconds (SD: 0.21 s), with the maximum latency of 1.89 seconds. Such a latency can be explained by the 1 Hz sensor polling rate and the Blynk server propagation delay that are acceptable in the context of parking management uses.

Switch off filter validations

This filter switch feature was evaluated by applying different combinations of slots occupancy, as well as switching the switch on and off. With a 100 percent accuracy in slot suppression/display logic, the dashboard accurately displayed vacancies in all 30 test scenarios, and therefore only vacant slots with a filter in place. Transition time of the filtered and unfiltered view in all cases was less than 500 Ms.

Comparison to Related Work.

Table III gives a comparison of the proposed system to those on the literature. Compared to other IoT-based parking systems, the system's performance is comparable in terms of cost, accuracy, and responsiveness [12], [13]. The suggested system is both competitive regarding accuracy and much cheaper to produce hardware-wise than image-processing-based systems, and it is also remotely accessible, a feature not found in simply local display systems.

The experimentative performance also showed the strength of the system in various situations such as distance and the object type. As it was discovered, there were only edge cases where sensor interference occurred with some inaccuracies that can be removed by proper placement and calibration. Generally, the system was functioning well and reliable in nature that established that the system is a feasible smart parking system.

VI. CONCLUSION

In this paper, we have introduced a smart parking system using ESP32 microcontroller and HC-SR04 ultrasonic sensors along with integration with the Blynk internet of things (IoT) cloud system. The system is able to accurately monitor Vacant/Occupied status slot-by-slot in real time on a remotely accessible dashboard with a detection accuracy of 98% and a response time of 1.34s on an average. The new filter option feature that enables users to put on a single toggle to view empty slots only is a convenient addition that minimizes the amount of cognitive load that a user with a vacant parking slot would have to bear to find available parking slots. The combination of low cost (estimated Bill of Materials not exceeding USD 15 to create four slot prototype), deployed easily, and wireless connection (Wi-Fi connectivity) features of the system makes it suitable in small-scale to medium-sized parking facilities including campus parking service, commercial complexes, and residential communities. Future research will involve multi-node ESP32 mesh networking of large-scale applications, and the addition of automated control of barriers at exit/entry points, and integrating machine learning models of predictive vacancy based on historic occupancy patterns.

The system offers a high potential of future good functionality and integration with the advanced technologies, in addition to the existing features it is availed with. Its functionality also can be extended with predictive analytics, automated reservation systems, and connect with the navigation services. This solution will also be applicable in construction of efficient and intelligent parking systems in smart cities in future since it is still under development.

ACKNOWLEDGEMENT

The authors would like to acknowledge the support of the Department of Computer Science and Engineering (IoT) at Raj Kumar GoelInstitute of Technology in the provision of laboratory facilities and infrastructure during the process of developing and testing this project.

REFERENCES

[1]. V. Paidi, H. Fleyeh, J. Håkansson, and R. Nyberg, “Smart parking sensors, technologies and applications for open parking lots: A review,” IET Intelligent Transport Systems, vol. 12, no. 8, pp. 735–741, Oct. 2018.

[2]. S. Srivastava, P. Sahana, and S. Dutta, “Smart parking system using IoT technology,” International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 5, pp. 227–230, May 2019.

[3]. A Patil, S. Kale, and S. Mane, “IoT based smart parking system using Arduino,” International Research Journal of Engineering and Technology (IRJET), vol. 6, no. 4, pp. 1200–1203, Apr. 2019.

[4]. K. Acharya and S. Kumar, “Smart parking system using Raspberry Pi and image processing,” International Journal of Computer Applications, vol. 180, no. 39, pp. 15–18, Apr. 2018.

ISSN: 2581-9429
Volume 6, Issue 1, May 2026
Impact Factor: 8.2

[5]. J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd, "Smart parking: An application of optical wireless sensor network," in Proc. Int. Symp. Applications and the Internet Workshops, 2007, pp. 66–69.
[6]. Espressif Systems, "ESP32 Series Datasheet," 2023. [Online]. Available: https://www.espressif.com
[7]. Blynk Inc., "Blynk IoT Platform Documentation," 2024. [Online]. Available: https://docs.blynk.io
[8]. A S. Malik and R. A. Khan, "Distance measurement using ultrasonic sensor and microcontroller," International Journal of Engineering Science and Computing, vol. 7, no. 4, pp. 10234–10237, 2017.
[9]. H. Shao, Y. Zhang, and X. Wang, "Smart parking system based on IoT technologies," in *Proc. IEEE Int. Conf. Internet of Things (iThings)*, 2016, pp. 104–109.
[10]. R. Yusnita, N. Fariza, and B. Norbaya, "Intelligent parking space detection system based on image processing," *International Journal of Innovation, Management and Technology*, vol. 3, no. 3, pp. 232–235, Jun. 2012.
[11]. M. Idris, Y. Leng, E. Tamil, N. Noor, and Z. Razak, "Car park system: A review of smart parking system and its technology," *Information Technology Journal*, vol. 8, no. 2, pp. 101–113, 2009.
[12]. S. Geng and J. Cassandras, "A new 'smart parking' system based on optimal resource allocation and reservations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1129–1139, Sept. 2013.
[13]. P. S. Chavan, P. R. Jadhav, and S. V. Patil, "IoT based smart parking system using NodeMCU," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 8, no. 2, pp. 45–48, Feb. 2019.
[14]. A Khanna and R. Anand, "IoT based smart parking system," in *Proc. IEEE Int. Conf. Internet of Things and Applications (IOTA)*, 2016, pp. 266–270.
[15]. S. Kumar and D. Singh, "Wireless sensor network based smart parking system," *International Journal of Computer Science and Information Technologies*, vol. 6, no. 2, pp. 1201–1204, 2015.

DOI: 10.48175/IJARSCT-34725